KTH Information and
Communication Technology

KTH Royal Institute of Technology

Dept. of Software and Computer Systems

Degree project in Information and Software Systems

Gaze-supported pointing input devices for day-to-day computer interaction

Author: Germán Leiva
Supervisor: Ralf Biedert

Examiner: Prof. Konrad Tollmar, KTH, Sweden

Abstract

In this work we assess four prototypes of gaze-supported input devices that aim to be more ergonomic alternatives than the mouse and touchpads for day to day computer usage. These prototypes (nunchuck, ring, smartphone and trackball) have three properties. First, they allow a distant interaction from the computer letting the user have 135-degree body-thigh sitting posture. Second, they allow a one handed interaction convenient for people with one healthy hand—even if the user has ergonomics complications in the other lower arm. Lastly, they are based on familiar input technologies and conventions that makes them easy to learn and use. An experiment was conducted to evaluate quantitative properties of the devices and user feedback during pointing and drag&drop. The smartphone ranked first in almost all the user feedback. The measures of speed and accuracy presented several outliers. In absolute terms the smartphone got the highest speed for pointing, the trackball got the highest speed for drag&drop and the smartphone got the highest accuracy (during pointing and drag&drop). The nunchuck ranked last in all the measures.

Contents

1 Introduction **3**
- 1.1 Background . . . 3
- 1.2 Problem . . . 4
- 1.3 Purpose . . . 4
- 1.4 Goal . . . 5
- 1.5 Methodology or Methods . . . 5
- 1.6 Delimitations . . . 5
- 1.7 Outline . . . 6

2 Theoretic Background **9**
- 2.1 Ergonomics . . . 9
  - 2.1.1 Neutral posture . . . 10
  - 2.1.2 Musculoskeletal disorders . . . 12
  - 2.1.3 Sitting posture . . . 13
- 2.2 Input devices . . . 14
  - 2.2.1 Indirect-Input devices . . . 14
- 2.3 Eye-trackers . . . 15
  - 2.3.1 Gaze-supported interaction . . . 19
  - 2.3.2 Pointing with your eyes . . . 20
  - 2.3.3 Gaze-supported input devices . . . 21

3 Prototyping a gaze-supported input device **23**
- 3.1 Design Space . . . 23
  - 3.1.1 User-Experience considerations . . . 24
  - 3.1.2 Ergonomics considerations . . . 24
- 3.2 Finding the candidates . . . 26
- 3.3 Architecture of the system . . . 28
- 3.4 Direct mapping . . . 29
- 3.5 Explicit versus implicit teleportation . . . 31
- 3.6 Candidates . . . 31
  - 3.6.1 Trackball . . . 31
  - 3.6.2 Joystick . . . 32
  - 3.6.3 Ring . . . 35
  - 3.6.4 Smartphone . . . 36

4 Evaluation **43**
4.1 Method 43
4.2 Procedure 44
4.3 Measures 45
4.3.1 Pointing 46
4.3.2 Drag and dropping 50
4.3.3 Overall Satisfaction 52

5 Conclusion **55**
5.1 Discussion 56

6 Further work **59**

List of Figures

2.1 Posture of an astronaut resting in zero gravity showing the neutral positions of the joints. The segment angles shown are means. Values in parentheses are standard deviations about the mean. The data was developed in Skylab studies and is based on the measurement of 12 subjects. Author: [20]. 11
2.2 The position of the hand at rest [22] 11
2.3 Standing desk. Copyright 2009-2014 Tinkering Monkey. 13
2.4 Remote systems. This thesis work exclusively with this type of eye-tracker 15
2.5 Head mounted system. 15
2.6 The pupil is the central transparent area (showing as black). The blue area surrounding it is the iris. The white outer area is the sclera. Author: Laitr Keiows. 16
2.7 The structure of the eye. Author: WebMD 17
2.8 A properly identified pupil (white cross) and corneal reflection (black cross) 17
2.9 The white cross is the true gaze position and the dots represent the measurements. The down left quadrant shows a low precision and low accuracy while the top right quadrant shows high precision and high accuracy. Based on a figure from [10, p. 34] 19
2.10 The IBM Track Point IV (red stick) was used in the MAGIC pointing experiment 20

3.1 A 135-degree angle increases the distance between the computer and the user 26
3.2 Handheld mouse 27
3.3 Game-controller 27
3.4 Ring Mouse 27
3.5 Smartphone 27
3.6 Custom input devices uses different protocols that a re-purposed mouse 28
3.7 The index finger controls the trigger while the thumb can be used to control the trackball or the scroll-wheel 32
3.8 A Wii controller composed by the nunchuck (left) and the Wii Remote (right) 32
3.9 The nunchuck is connected to the Arduino board 33

3.10 The first mappings on the nunchuck before leaving the explicit teleportation approach . . . 33
3.11 The four concentric regions of the analog stick . . . 34
3.12 The ring mounted on the user's index finger . . . 35
3.13 The mapping of the ring . . . 36
3.14 States sensed by a mouse versus states sensed by touch-operated devices such as smartphones. Adapted from (Hinckley, Czerwinski & Sinclair, 1998a) . . . 37
3.15 A MacBook Pro manufactured by Apple showing the "trackpad" a multi-touchpad used as the pointing device of the machine . . . 38
3.16 Reach of the thumb while holding a device with one hand [28] . . . 39
3.17 The layout and the mappings of the smartphone prototype with their corresponding mappings . . . 39
3.18 In order to implement "tap and drag" two timer states were introduced 40
3.19 We disabled the landscape mode that allowed a keyboard input to focus our study in pointing tasks . . . 42

4.1 A target is represented as a white circle . . . 44
4.2 When the cursor is on top of the target, the target is highlighted . . 44
4.3 *Draggable object* as a circle and *drop area* as a dotted circle . . . 45
4.4 The *drop area* is highlighted when the target is on top of it . . . 45
4.5 Movement time (ms) data represented in a boxplot for all the sizes and distances combination for the pointing task. Dots and stars are outliers . . . 45
4.6 Movement time (ms) data represented in a boxplot for all the sizes and distances combination for the drag and dropping task. Dots and stars are outliers . . . 46
4.7 Pointing speed perceived by the participants for each device (the higher the better) . . . 46
4.8 Comparison of mean movement time (in ms, the lower the better) during the pointing task across all the devices . . . 47
4.9 Mean movement time (ms) of the mouse during the pointing task showing all the sizes and distances combinations . . . 48
4.10 Mean movement time (ms) of the smartphone during the pointing task showing all the sizes and distances combinations . . . 48
4.11 Pointing precision perceived by the participants for each device (the higher the better) . . . 49
4.12 Comparison of mean distance from cursor to the center of the target (in pixels) during the pointing task across all the devices. Low values mean higher accuracy . . . 49
4.13 Comparison of mean error count during the pointing task across all the devices . . . 49
4.14 Dropping speed perceived by the participants for each device (the higher the better) . . . 50

4.15 Comparison of mean movement time (in ms, the lower the better) during the drag and drop task across all the devices 50
4.16 Drag and drop precision perceived by the participants for each device (the higher the better) 51
4.17 Comparison of mean distance from cursor to the center of the target (in pixels) during the drag and drop task across all the devices. Low values mean higher accuracy 51
4.18 Comparison of mean error count during the drag and drop task across all the devices 52
4.19 Ease of learning 52
4.20 Ease of use 53
4.21 Comfortability 53
4.22 Overall satisfaction 54

5.1 Final device ranking for perceived and measured values 55
5.2 A participant that is using the desk to hold the smartphone 56
5.3 A participant using the desk to support his arm while using the nunchuck 57
5.4 A participant dragging with their non-dominant hand and moving with their dominant hand while using the smartphone 57
5.5 A participant supporting the trackball with his leg while hold the device with two hands 58
5.6 A participant supporting their dominant arm with their non dominant arm while using the trackball 58

6.1 An example of how the input devices can handle text input and not only pointing tasks 60
6.2 “Sticky textures”, an example of how we can add haptic feedback to the smartphone’s screen 61

Chapter 1

1 Introduction

In this work we assess four prototypes of gaze-supported input devices that aim to be more ergonomic alternatives than the mouse and touchpads for day to day computer usage. These prototypes (nunchuck, ring, smartphone and trackball) have three properties. First, they allow a distant interaction from the computer letting the user have 135-degree body-thigh sitting posture. Second, they allow a one handed interaction convenient for people with one healthy hand—even if the user has ergonomics complications in the other lower arm. Lastly, they are based on familiar input technologies and conventions that makes them easy to learn and use. An experiment was conducted to evaluate quantitative properties of the devices and user feedback during pointing and drag&drop. The smartphone ranked first in almost all the user feedback. The measures of speed and accuracy presented several outliers. In absolute terms the smartphone got the highest speed for pointing, the trackball got the highest speed for drag&drop and the smartphone got the highest accuracy (during pointing and drag&drop). The nunchuck ranked last in all the measures.

1.1 Background

This thesis is focused on the design considerations that an eye-tracker's companion device should take in order to provide means for an ergonomic day to day interaction between a user and a computer (laptop or desktop environment). The main goal of this companion device is to allow a better ergonomic usage than the most common pointing input devices (i.e. mouse).

The main field of study that covers this thesis is Human-Computer Interaction (HCI).

The increase in workplace computer usage since 1980 has been accompanied by a corresponding increase in incidence rates of chronic musculoskeletal disorders reported [18]. We believe that a pointing device that works in conjunction with an eye tracker can reduce the repetition and duration of the physical movements on the most common pointing tasks. The stage of this work is prototypical so our

main objective is not to implement the final pointing device but to outline the design considerations that such device should address and the most suitable input technologies in order to provide a more ergonomic usage.

Our starting hypothesis is that by using gaze information in combination with manual inputs we can reduce the physical effort needed in order to move the cursor on a computer screen.

1.2 Problem

While eye-tracking technologies were used primary as a research method, the same technologies can be used as a new channel for interaction. This type of interaction based on gaze information is called *gaze-interaction*. Due to the natural tendency of first looking and then pointing [13], *gaze-interaction* can use the gaze information as an input for pointing tasks. While pointing appears to be a simple usage of the gaze information, reality has proven that it is harder than it seems. The quality of the measurements obtained by the eye tracker are not optimal and are coupled with the combined effects of eye-tracker specific properties (such as sampling frequency, latency, and precision), participant-specific properties (such as glasses and mascara), inconsistencies during calibration and in the filters used [10, p. 29].

We want to improve the ergonomic physical conditions of the desktop environment by combining the typical mouse tasks with eye tracking technologies.

Our drive is to answer the question: which pointing input technology provides the highest productivity and the highest comfort when is used in combination with eye-tracking technologies?

1.3 Purpose

This thesis describes the work with designing a prototypical ergonomic alternative to the mouse for the day to day computer usage based on eye tracker technologies. We are calling this prototype a “companion device”.

The purpose of this companion device is to allow a better sitting posture, a reduction in the repetition and a reduction in the duration of the pointing task.

1.4 Goal

The goal of the work is to evaluate if these prototypical companion devices could provide physical ergonomics benefits and, at the same time, they provide a familiar and productive usage of the computer.

In this work we are prototyping four candidates eye-tracker's companion devices in order to answer the following questions:

- Which prototype provides the most positive *overall satisfaction*?
- Which prototype provides the highest *productivity* (in terms of low movement time and high accuracy)?
- Which prototype provides the best *ergonomic experience*?

The main benefit of this work is to provide an alternative to the mouse for people with ergonomics complications or people that want to avoid such complications.

1.5 Methodology or Methods

In order to evaluate our approach, we prototyped four eye tracker's companion devices based on existing input technologies that are already available in the market (some of them are already used as pointing devices and others were re-purposed for this study).

We designed the mappings between these physical devices and the most used mouse inputs (i.e. move, button down and scroll) based on user testings and preliminary experiments.

After that, we conducted an experiment to evaluate not only measures related with the productivity of the prototype but also user feedback related with the ergonomics characteristics of the device.

1.6 Delimitations

This work is focused on pointing tasks. While day-to-day computer interaction includes other types of inputs (e.g. text inputs with a keyboard) we decided to focus our work on pointing due to the clear benefits that gaze information has with that type of task. In the future work the companion device can be extended to include some type of text input.

The prototypes are based in existing input devices that are modified or re-purposed to work with the eye-tracker but none of them was created during this work. In other words, industrial design considerations were not part of this study.

In order to evaluate the ergonomics capabilities of the companion devices we are using questionnaires. We are not measuring any muscular strain using EMG readings (electromyography).

1.7 Outline

The thesis describes the work with designing four prototypes and evaluating them.

Recent advancements in eye tracking technology, specifically the availability of cheaper, faster, more accurate and easier to use trackers have inspired increased eye movement and eye tracking research efforts [7]. Nowadays, eye trackers are not only used for scientific research, they are a valuable tool for commercial testing (e.g. user experience testing or market research) and assistive technologies (i.e. eye control for people with disabilities).

Cost reduction of manufacturing eye-trackers will make possible to introduce this technology to a broader market and integrate them to consumer products like laptops and desktop environments allowing users to control —at some extent— their computer's interface with their eyes. Even if the technology is available constructing eye tracker interfaces have a big challenge, people are not accustomed to operate devices simply by moving their eyes [14].

We believe that an interesting introduction of the eye tracking technologies for this broader market are the potential benefits that a gaze-interaction can deliver to a computer user that is suffering ergonomics complications or for users that want to prevent the development of such complications. Today, the dominants input devices for laptop and desktop computers are the mouse and the keyboard, both have associated ergonomics problems generated by the bad sitting posture and the improper angle of the arms and hands joints while using them, further exacerbated by the repetitive and continuous usage.

In this thesis we explore the design of an eye-tracker's companion device as an alternative input device to the mouse and a complementary input channel of the gaze data collected with the eye-tracker for the day to day usage of a laptop or desktop environment.

Our main objectives with this companion device are to:

- Reduce the amount of physical movement that the user's hands should exert
- Perform as good as the mouse in terms of provided capabilities and time performance (i.e. pointing, selection and activation)

- Allow a distant interaction from the screen (i.e. without the user's hand on top of the keyboard or mouse)

Chapter 2
2 Theoretic Background

This thesis was conducted in collaboration with the research and development department at Tobii Technologies AB in Sweden, during the months of January and June of 2014.

While the field of Ergonomics/Human Factors and Human-Computer Interaction are big and relevant today, there is little or no research in the study of *gaze-supported interaction* from the point of view of the ergonomics capabilities that it could provide. Some authors mention the potential physical ergonomics benefits [3, 12, 30, 24] but this is not the main focus of their research. We believe that eye-tracking can be used as a way to improve the ergonomic usage of the input devices that controls the still dominant WIMP (Windows Icons Menus and Pointers) interfaces of the desktop computers.

In this chapter we are giving a brief introduction to Ergonomics and the areas relevant to this study, the state of the art in mainstream computer input technologies and, finally, the state of the art of eye-tracking technologies as a complementary input channel for pointing and selection tasks.

2.1 Ergonomics

In Europe, the field of ergonomics was inaugurated after the Second World War; even if Jastrzebowski produced a philosophical treatise on *Ergonomics: The Science of Work* in 1857, it seems to have remained unknown outside Poland until recently [5]. In the United States a similar discipline emerged, it is known as human factors but its scientific roots were grounded in psychology instead of biological sciences. Nowadays, the Human Factors Society in the United States has changed its name to the Human Factors and Ergonomics Society indicating that the two areas have a big overlapping. The Human-Computer Interaction field was heavily influenced by Human Factors and Ergonomics; for an overview of the evolution of the field, please refer to the Introduction in [11, p. xxvii].

According to [5, p. 1] ergonomics is "the study of the interaction between people and machines", as well as the factors affecting this interaction, with the "purpose [to] improve the performance of systems by improving the human machine interaction". This can be achieved by *designing-in* factors that facilitate interaction, or *designing-out* factors that impede it. Problem domains are, amongst others, efficiency improvements, reduction of fatigue, or the prevention of accidents, injuries and errors. The field of Ergonomics it is huge; it considers all the elements and their interactions in a human-machine system (i.e. the human, the machine and the environment) but in this thesis, we are focusing on the physical ergonomics considerations for the human that is using a computer system like a desktop or a laptop environment.

The increase in workplace computer usage since 1980 has been accompanied by a corresponding increase in incidence rates of chronic musculoskeletal[1] disorders reported [18]. Musculoskeletal disorders occurs among others when there is a mismatch between the physical requirements of the job and the physical capacity of the human body. The disorders occurs when the body part is called on to work harder, stretch farther, impact more directly or other function at a greater level then it is prepared for [21].

2.1.1 Neutral posture

One of the most important considerations for avoiding these disorders is to let the user maintain, as much as possible, a *neutral posture* while executing a task. Neutral posture refers to the resting position of each joint, is the position in which there is the least tension or pressure on nerves, tendons, muscles and bones (Figure 2.1).

In a neutral posture the muscles are at their resting length, neither contracted nor stretched and they can develop maximum force most efficiently. One aspect of ergonomic redesign is the reworking of tools, work stations and processes to allow the worker's joints to remain in neutral position as much as possible. The movements of the different joints are described with a particular terminology, covering all of them is out of the scope of this thesis but we are providing a brief explanation [27] of some of the body parts with their respective movement names in parenthesis:

- **Fingers** are gently curved and they are not spread apart (Figure 2.2). They are neither fully straightened out (extended) nor tightly curled (flexed).
- **Wrist** is in line with the forearm. It is neither bent up (extension) nor bent down (flexion). It is not bent towards the thumb (radial deviation) nor towards the little finger (ulnar deviation).

[1] The musculoskeletal system gives human the ability to move using the muscular and skeletal system.

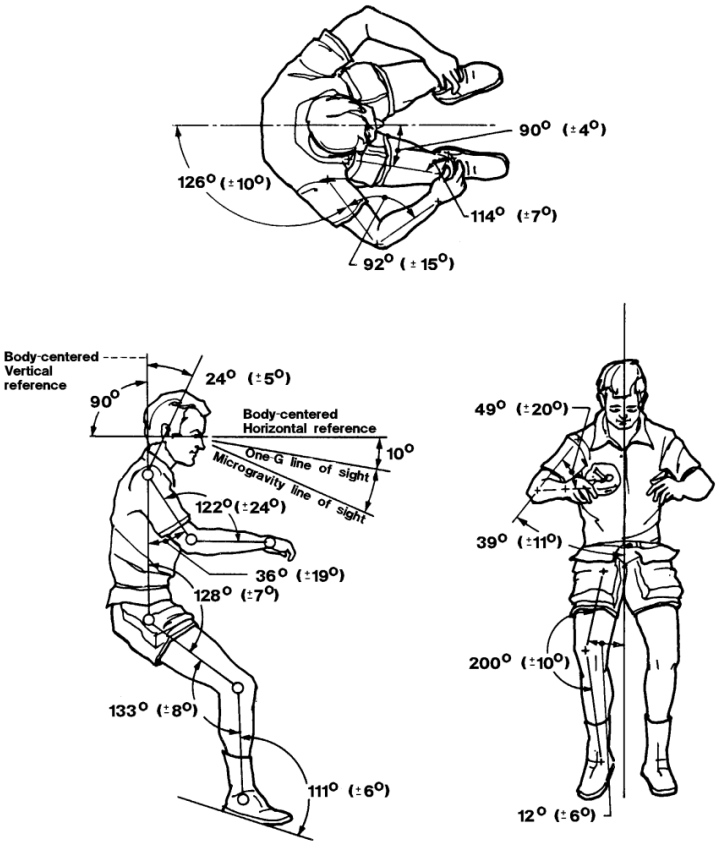

Figure 2.1: Posture of an astronaut resting in zero gravity showing the neutral positions of the joints. The segment angles shown are means. Values in parentheses are standard deviations about the mean. The data was developed in Skylab studies and is based on the measurement of 12 subjects. Author: [20].

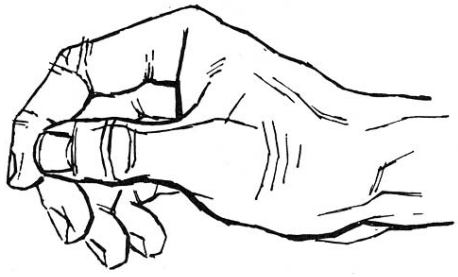

Figure 2.2: The position of the hand at rest [22]

- **Forearm** rests with the thumb up. It is not rotated to make the palm face down (pronation) or up (supination).
- **Elbow** is in a neutral position when the angle between forearm and upper arm is close to a right angle (90 degrees). Some extension, up to 110 degrees, may be desirable.
- **Upper Arm** hangs straight down. It is not elevated to the side (abduction), pulled across in front of the body (adduction), raised to the front (flexion) nor raised towards the back (extension).
- **Shoulder** are neither hunched up nor pulled down, and not pulled forward or back.
- **Neck**, the head is balanced on the spinal column. It is not tilted forward, back or to either side. It is not rotated to the left or right.
- **Back**, the spine naturally assumes an S-shaped curve. The upper spine (i.e. thoracic region) is bent gently out; the lower spine (i.e. lumbar region) is bent gently in. These bends are called kyphosis and lordosis, respectively. The spine is not rotated or twisted to the left or right, and it is not bent to the left or right. Whether standing or sitting, the trunk does not bend forward (flexion) or backward (extension) by much; although a good backrest on a seat does allow extension.
- **Lower Body**, under conditions of weightlessness (Figure 2.1), the lower body naturally assumes a neutral, fetal position; hip and knee joints are somewhat bent.

2.1.2 Musculoskeletal disorders

Many people spend a significant amount of time using computers for entertainment or work purposes. Repetitive strain injuries (RSI), such as carpal tunnel syndrome and tendonitis, have been seen in adults who work with computers and are also increasing in number among college age and younger students [29].

There have been objections regarding the existence of RSI as a separate medical condition [6]. Clear incidence and prevalence rates for RSI and other work related disorders are hard to find when up to 9% of a population have suffered pain or discomfort in the arm [6]. According to a study of 2010 in Europe [15], musculoskeletal disorders in the hand/wrists are reported in between 20% and 50% of the cases, in shoulders 16% to 38%, in elbows 5% to 15% of the cases. In the same study 8% of workers missed up to one week of work due to WMSDs (Work-Related Musculoskeletal Disorders), and 2% up to 30 days. With this numbers we can see that disorders in the hand/wrists are the most reported ones.

The general factors [5, p. 170] leading to WMSD problems are known to be:

- Force required for an operation
- Posture of limbs, and deviations from resting positions
- Repetition of tasks, especially in high volume
- Duration of tasks
- Stress and anxiety

2.1.3 Sitting posture

Human are designed to stand on two legs, but they are not designed to stand still [5, p. 115]. A 2010 study [25] examining 43 papers related to occupational sitting found "limited evidence" supporting increased health risks or mortality risk due to occupational sitting, with 5 papers concluding the opposite.

In general, both seated and standing postures involve deviations from neutral posture. The usage of standing desk (Figure 2.3) is increasing and allows the user to interact with the computer while standing, nevertheless, sitting remains as the prefer posture when using a desktop environment. For this reason, we are focusing this thesis only on the sitting position.

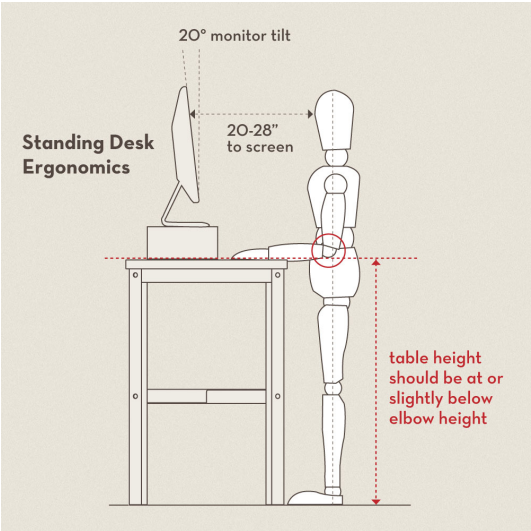

Figure 2.3: Standing desk. Copyright 2009-2014 Tinkering Monkey.

2.2 Input devices

An input device is a transducer that senses physical properties of people, places, or things. What we normally conceive of as an input device (such as a mouse) is often a collection of transducers, such as a relative *(x,y)* motion sensor, physical buttons, and a wheel for scrolling [11, p. 100].

One big advantage of the indirect input devices is that they are decouple from the output, this allows to use an indirect input device to control an output (e.g. the screen) that is distant from the user and the input device.

2.2.1 Indirect-Input devices

An indirect input device is one that does not provide input in the same physical space as the output. Indirect devices eliminate occlusion of the screen by the user's hand and fingers. However that require more explicit feedback and representation of the input device (such as a cursor), intended target on the screen (such as highlighting icons when the cursor hovers over them), and current state of the device (such as whether a button is held or not) [11, p. 104].

Current indirect input device technologies can be categorized in:

- Touchpads
- Multitouch pads
- Mice
- Trackball
- Isotonic joystick
- Isometric joystick

These are mainstream technologies and we are not explaining in detail each one of them, for an in deep explanation of these technologies refer to [11].

An isotonic joystick is a handle stick where the user can freely change the position of it, in contrast on an isometric joystick the position of the stick remains more or less constant. In this work we are working with isotonic joysticks due to their increased popularity over isometric joysticks; from now on we are referring to this category only as "joysticks".

2.3 Eye-trackers

Eye-trackers make possible the use of eye-movement data. Eye-tracker technology has been around since the late 1800s but they were technically difficult to build, mostly mechanical, and not very comfortable for the participants [10, p.9].

Eye tracking technologies are gone a long way and recent advancements, specifically the availability to build cheaper, faster, more accurate and easier to use trackers have increased the interest in this field [7].

In the current landscape eye-tracking systems can be divided into two big groups, remote and head-mounted systems. Each type of system has its respective advantages. For example, remote systems (Figure 2.4) are non-intrusive because they are statically installed near the computer screen without the need of attaching any device to the user itself, but at the same time are less accurate due to the fact that the position of the user's eyes is not fixed in relation with the tracker. This type of eye-trackers allows some head movement as long as the head of the user is located inside the *headbox*[2] of the remote system. Even if the remote system allows some head movement is not as flexible as a head-mounted systems (Figure 2.5) that is integrated in the frame of a pair of glasses and allows total freedom in terms of head and body movements.

Figure 2.4: Remote systems. This thesis work exclusively with this type of eye-tracker

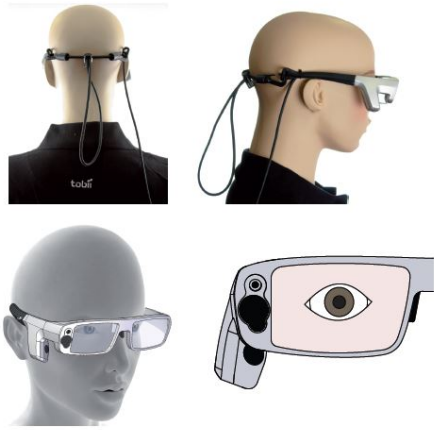

Figure 2.5: Head mounted system.

Even if the presence of the remote eye-tracker is noticeable, it is not as intrusive as the head-mounted system. Commonly, the remote system can be integrated with the

[2]The headbox is the volume relative to the eye-tracker in which the user can move without compromising the quality of tracked eye data. Towards the edge of this volume the data will be gradually poorer.

computer environment over a standard port (e.g. USB) and after the installation of the appropriate software for the target operating system it provides real-time gaze data; in contrast not all head-mounted system provides real time gaze data and their main goal is to provide mobility to the user to record his gaze data in order to analyze it later on. For these reasons in this thesis we are working exclusively with remote-eye trackers.

Tracking the eye movements has as a main usage estimating accurately where someone looks on a stimilus (e.g. the computer screen), we will refer to this position as the *point of gaze*.

The dominant method for estimating the point of gaze from an image of the eye is based on the pupil and cornea reflection tracking [10, p.24]. The pupil (Figure 2.6) is a hole located in the center of the iris that allows light to enter the retina; the cornea is a transparent spherical membrane that covers the front part of the eye, both can be located in Figure 2.7. To recognize these elements you can think of the cornea as the part of the eye where you usually see reflections and the pupil as the region of the eye that turns red (red-eye effect) when someone's take a close-up picture using flash in a low light environment.

Figure 2.6: The pupil is the central transparent area (showing as black). The blue area surrounding it is the iris. The white outer area is the sclera. Author: Laitr Keiows.

As a general and simple overview, the eye tracker that uses the pupil and cornea reflection method should have at least one video camera in order to obtain the images of the eyes and an infrared light to generate the reflection in the cornea. After the image has been acquired, it is analyzed to detect the position of the pupil (a dark oval in Figure 2.8) and the corneal reflection (a smaller bright dot Figure 2.8). Geometrical calculations combined with a calibration procedure are finally used to map the positions of the pupil and corneal reflection to the data sample *(x,y)* on the stimulus [10, p.24].

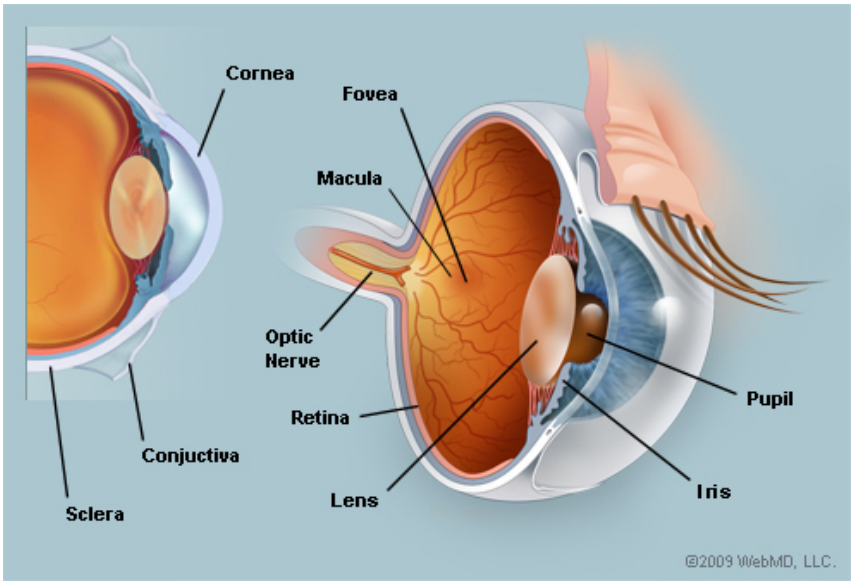

Figure 2.7: The structure of the eye. Author: WebMD

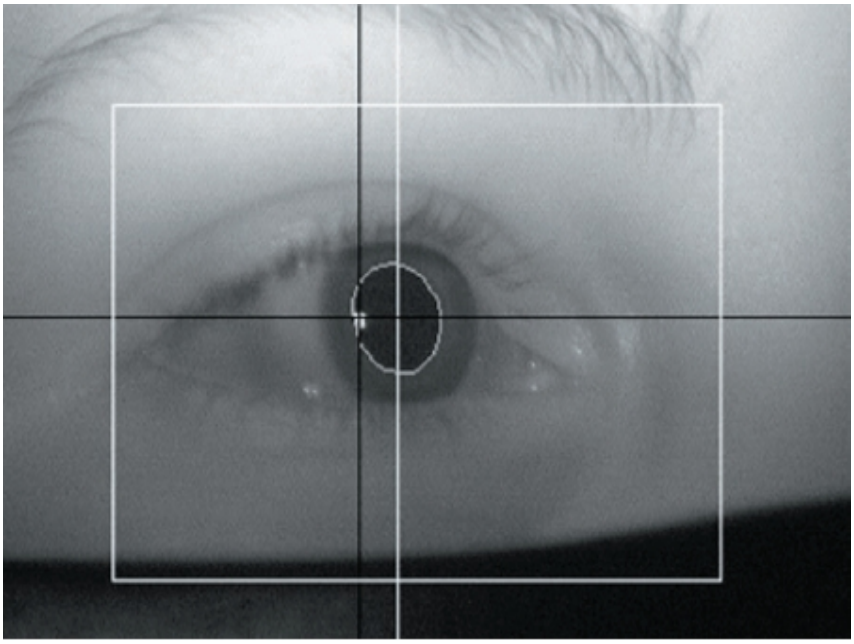

Figure 2.8: A properly identified pupil (white cross) and corneal reflection (black cross)

This method sounds fairly simple but it has some weaknesses. Eyelid occlusion of the pupil may cause *offsets* (incorrectly measured gaze positions), also increased imprecisions in the data in some parts of the visual field could occur and extreme gaze angles may cause the reflection detection to get lost [10, p.27]. While some of these problems can be mitigated with better algorithms and models, imprecisions and inaccuracies cannot be totally eliminated.

We are introducing the naming of the most common movements of the eye in order to understand the potential problems that they can generate while we try to map the point of gaze to the corresponding (x,y) coordinate on the stimulus.

The most relevant movement of the eye for the eye tracker is not a movement per se, is the absent of significant movement for a particular period of time. This is called a *fixation* and last anywhere from some tens of milliseconds up to more than a second (250 milliseconds on average while reading silently). One complication when we try to measure fixations is that the eye is not completely still, there are three types of micro-movements called *tremor*, *microssaccades*, and *drifts*. Tremor is a small movement whose exact role is unclear (possibly due to a imprecise muscle control) and has a frequency of approximately 90 Hz. Drifts are slow movements taking the eye away from the center of fixation, and microsaccades make possible that the eye returns to its original position in a fast way [10, p.22]. The motion from one fixation to another is called *saccade*, they are fast (between 30 and 80 ms) and is safe to consider the user "blind" during a big part of this movement. Another eye movement is the *smooth pursuit*, saccades can be performed in any stimulus but during a smooth pursuit the eyes requires something to follow.

These eye movements are relevant in order to understanding the inaccuracies and imprecisions of the measures illustrated in Figure 2.9.

Some concepts related with accuracy and precision are:

- **Offset** a distance between the calculated fixation and the intended target (e.g. top left quadrant of Figure 2.9).
- **Drift** a gradually increasing offset.
- **Oculomotor noise** often called *jitter*, refers to the fixational eye movements tremor, microsaccades and drift.
- **Enviromental noise** variation in the gaze position signal caused by external mechanical and electromagnetic disturbances in the environment.
- **General noise** the combination of oculomotor and environmental noise.

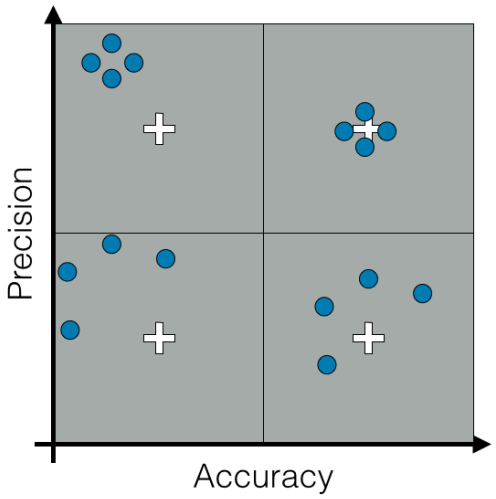

Figure 2.9: The white cross is the true gaze position and the dots represent the measurements. The down left quadrant shows a low precision and low accuracy while the top right quadrant shows high precision and high accuracy. Based on a figure from [10, p. 34]

2.3.1 Gaze-supported interaction

Generally, fixations can be considered as a point of attention for the user and because of that, the first usages of eye tracking technologies can be found in scientific research.

Nowadays, eye trackers are not only used for scientific research and they are considered a valuable tool for commercial testing (e.g. user experience testing or market research) and assistive technologies, this is the usage of computers by people with disabilities (i.e. mobility impairments) using only gaze information. Users that cannot interact with a computer using their hands in order to operate peripherals like the mouse and the keyboard can use eye trackers to create a new input channel with the computer.

As observed in [26] one of the most frequent operations which occurs in human computer interaction is the selection of an object displayed on the screen, and one of the principal ways in which a human observer directs his or her visual attention to objects in the immediate environment is by fixating them. Combining these two observations, *gaze only interaction* allows the user to activate widgets like buttons by fixating them. Only considering fixations to generate activations in the computer has a general problem that is known as "Midas Touch"[3].

[3] Also known as the Golden touch, King Midas is popularly remembered in Greek mythology for his ability to turn everything he touched with his hand into gold

In general, people are not accustomed to operate devices simply by moving their eyes, they expect to be able to look at an item without having the look "mean" something [14]. The ideal would be to activate something by the user's gaze, only when he wants to and not generate any interaction when he does not have that intention. There are two common methods to solve this problem: one is to use dwelling times, when a fixation last more than a predefined time the interaction with the fixated object will be triggered; another is to use an extra input different from the gaze information to trigger the activation (e.g pushing a physical button).

2.3.2 Pointing with your eyes

A really interesting area for the use of gaze information is pointing because people tend to look at the object they wish to interact with [13]. The idea of only pointing with your eyes makes sense because it is a very fast pointing technique compared with other pointing devices [23] and eye movements offer near fatigue-free pointing [2]. But, as explained before using only your eyes to point is a difficult task mainly due to the inaccuracies of the gaze information provided by the eye-tracker and how the user triggers actions after pointing. While dwelling times is one technique used to overcome Midas Touch it adds a delay between the intention and the action, another approach that is proven to be faster is to use the eye movements as a supporting interaction channel in combination with other input modalities, such as speech, hand and body gestures, and other input devices; [24] call this type of interaction *gaze-supported interaction*.

This idea of *gaze-supported interaction* was introduced in [30], the authors mention different input devices that can be suitable for the experiment such as a standard mouse, a touchpad and a stick (a miniature isometric joystick); the mouse was discarded because of its tendency of been off of the pad and only the stick (Figure 2.10) was used in the experiment.

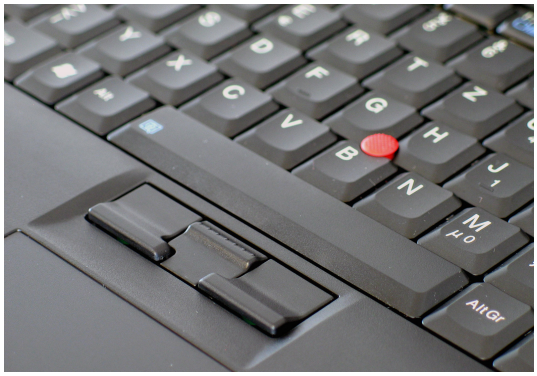

Figure 2.10: The IBM Track Point IV (red stick) was used in the MAGIC pointing experiment

Two MAGIC [4] pointing techniques where implemented in [30]. The first technique was called "liberal", with this technique each new fixation outside a boundary (a certain distance from the current gaze point) teleports the cursor to the new fixation. The second technique was called "conservative" and only teleports the cursor if the manual input device has been actuated. While the "liberal" approach was found faster in the experiment and overall subjects liked it better for its responsiveness, in a real life computer setting the over-activeness of the cursor (i.e. teleporting all the time to the current fixation event without triggering the manual input device) makes it an unusable approach. An interesting add-on on the "conservative" technique was the usage of an intelligent biased, instead of teleporting the cursor to center of the gaze area the cursor position is offset to the intersection of the manual actuation vector and the boundary of the gaze area but results does not shown a significant improvement by using the intelligent bias.

2.3.3 Gaze-supported input devices

Studies that compare input devices and their performance in common computer task like pointing and dragging can be found [19] in the literature. Finding studies that compare the performance of input devices that are augmented with eye-tracking capabilities is difficult.

While [24] proposed several interaction techniques that can be implemented in a hand-held multitouch device (an iPod touch in their experiment) they do not compare their performance with a mainstream pointing device like the mouse. From the three types of interaction techniques proposed only two (MAGIC touch and zoom lenses) are target agnostic (i.e. the pointing is decoupled of the knowledge of the position of the potential targets). We discarded the zoom lenses approach (a magnifying lens appears near the point of gaze making the refinement of the cursor position easier) because it incorporates an intermediate state that increases the time of the interaction. The technique refereed as MAGIC tab cycles between the targets that are within a selection mask based on the gaze point, in order to implement this technique knowledge about the targets needs to be provided making the technique target aware instead of target agnostic (MAGIC tab works similar to the TAB key of the keyboard). While more and more operating systems provides ways to expose and manipulate the widgets of other applications a target agnostic approach makes the interaction technique suitable for any application without the need to expose or modify anything. For this reason in this study we are only focusing on pointing techniques that are target agnostic.

Our main contribution with this work is to compare different input technologies when they are combined with the gaze information.

[4] Manual And Gaze Input Cascaded or Manual Acquisition with Gaze Initiated Cursor

Chapter 3

3 Prototyping a gaze-supported input device

3.1 Design Space

We want to design, prototype and evaluate an input device that works in combination with the gaze information collected from an eye-tracker; we are calling this device "companion device" (i.e. an eye-tracker's companion input device).

The objective of this companion device is to make possible the usage of the computer by people with ergonomics problems in one of their lower arms (e.g. fingers, wrist, forearm). The device should be also usable by people that does not have ergonomics complications. This companion device should show a comparable productivity while pointing in terms of speed and precision with the traditional input devices (i.e. mouse and touchpad).

Humans tends to look to the object of interest before interacting with it. On WIMP interfaces this notion can be translated to: users look to a widget of interest (e.g. a button) before moving the cursor to interact with that widget. This rule of first looking and then interacting generally applies but there are exceptions; according to an study run in the wild [16] eye leads the mouse click about two thirds of the time. After getting used to a particular WIMP interface, users do not necessarily look to the options before moving their hands. Lets take for example the case of a menu; users could move their hand even before the options are displayed, due to the repetitive usage of the application the user knows beforehand that the desired option will be placed in a particular position even without looking at the screen. Even if there are exceptions to the rule of "first look, then interact" we are building the companion device on top of it because is the most common case while pointing.

We are taking the gaze information to move the cursor near to the gaze point as soon as the intention of interacting is detected on the companion device. The rapid

movement of the cursor from his current position to a position near the gaze point will be referred as a "teleportation". We say near the gaze point and not exactly on the gaze point because of the inherent inaccuracies that the eye-tracker has. After this teleportation, the companion device is used to correct the position of the cursor to the desire position and trigger an action there. This gaze-supported interaction is similar to the MAGIC pointing technique presented in [30].

The are many considerations to take into account on the design of the companion device; it needs to be easy to learn and use, it needs to allow a distant interaction with the computer, it should reduce the force and repetition of the physical movements needed to perform the most common computer tasks and also be usable with only one hand.

3.1.1 User-Experience considerations

Eye-tracking technologies are not new but neither mainstream. Tobii wants to bring this technology to as many people as possible, and for that reason the companion device needs to be easy to learn. Getting used to the eye-tracker alone (to the use of the gaze as new way to control the computer) requires some effort from the user. On top of getting used to the eye-trackers, users needs to learn how to use the gaze in combination with the companion device. This put the ease of learning and use as two relevant factors to measure during the experiment.

In order to achieve this ease of learning we decided to use input technologies that are already out in the market and are considered mainstream. We are describing more in detail the candidates in the next section.

Requirements like installation and configuration where considered but are not the main focus of this work.

3.1.2 Ergonomics considerations

The usage of personal computers can generate musculoskeletal disorders. Some of the factors related with these disorders were presented in the first chapter and included:

- Posture of limbs, and deviations from resting positions
- Repetition of tasks, especially in high volume
- Duration of tasks

The companion device should address these factors in order to minimize or even prevent these complications.

The most commonly used position while using desktop and laptop computers still is the sitting position. The way the user sits has an important effect on the overall posture of limbs and deviations from resting position. While some ergonomics experts recommend a 90-degree angle between the thighs and the body, a 135-degree body-thigh sitting posture (Figure 3.1) has been demonstrated to be the best biomechanical sitting position [1]—this means that 135-degree is the angle that minimizes the strain on the lower back. We cannot question in deep the results of [1] but the size of the sample used (20 participants) seems small to conclude one and only one angle number that should fit all the anthropomorphic variations that potential users can have. Leaving aside our skepticism, lets consider this sitting angle of 135-degree the one that we want for our users. Current computer setups requires users to be at a close or mid-range distance from the support of the computer (e.g. the desk or the case of the laptop, should be positioned at a distant not bigger than the length of the arms fully extended) in order to reach the pointing input device (e.g. the mouse or the touchpad).

Interacting with the arms fully extended increase the variance with the neutral position of the arms, so the current computers setups are not compatible with a 135-degree angle for sitting. Because of that the companion device needs to be located close to the body of the user instead of near the computer. Three options appeared in the creation of our design space: the companion device could be mounted on the user's body, held by the user or mounted to the furniture (e.g. the arm-chair). While the last option has the benefit of removing all the weight lifting from the user's body (i.e. reducing the force of the task) and in consequence improving the ergonomics conditions, the setup is not familiar for the user and requires not only the creation of an input device but also a complete furniture or mechanism to mount the device to it. In this work we decided to use the first two approaches, the device is either held by the user or mounted on the user, [17] called these type of devices *elevated devices*.

To reduce the repetition of movements and also the duration of the task, we decided to combine the companion device to the gaze information provided by the eye-tracker. We are taking this gaze information to move the cursor near to the gaze point as soon as the interaction intention is detected on the companion device. The footprint generated by the user using a traditional input device without the help of gaze information should be greater if we compare it to the footprint generated while using the companion device. This fact is not demonstrated in this work but a simple observation can support the idea. Imagine a scenario where the cursor is in the top left corner of the screen; if the user wants to move the cursor to the bottom right corner of the screen using a traditional input device the amount of movement that the lower arm needs to do is clearly bigger when is compared with a cursor teleportation using gaze information. This teleportation approach reduces the overall

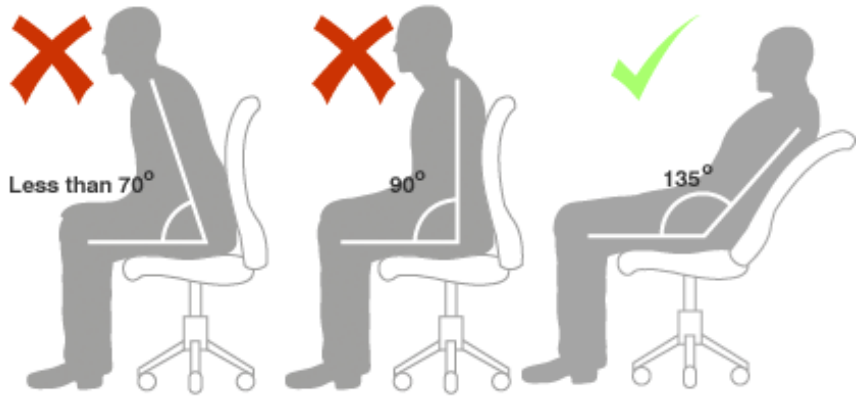

Figure 3.1: A 135-degree angle increases the distance between the computer and the user

duration of the task and also the amount of repetitive movements that the lower arms do.

In terms of the overall duration of the task, we are not aiming to improve the productivity achieved with the companion device if we compared it with the one that you get when using a traditional input device. Our goal is to reach a productivity comparable with the one achieved by traditional input devices but not lesser. This is highly related with the use of menus and direct mappings explained in the following section.

Having a companion device capable of being used with only one hand was another consideration to take. Our first hypothesis was that people with ergonomics problems in one hand should be able to use the device with their other healthy hand. This device cannot be used by people with ergonomics problems in both hands or that have injuries severe enough to make them unable to hold the device or push a button. There are other devices and forms of interactions for people with those types of motor disabilities in both hands and they are not part of this work.

3.2 Finding the candidates

In order to evaluate the potential of this companion device we decided to prototype four representatives candidates of the major indirect input technologies. We acknowledge that one candidate does not fully explore all the possible form factors and design decisions behind the indirect input technology that is representing, but we believe that it is a good starting point.

Direct input technologies where the output (e.g. the screen) and the input are together are discarded because they do not provide means to have a distant interaction with the computer.

Image-based motion sensing input devices that does not require the user's contact like the LeapMotion or the Microsoft Kinectare not considered in this work based on the assumption that a device that forces the user to maintain their arms in an special position for a long period of time is not reducing the force nor helping to reach a neutral position. Another consequence of this type of technologies is that any haptic feedback is lost, there are no buttons to press or device to hold; this makes them not suitable for tasks that requires precise pointing over long period of interaction.

For these reasons, we are focusing this work on indirect input technologies. Because we want to have a small learning curve for people transitioning from the traditional input device to this companion device, we based our prototypes in devices that are already available on the market and were used as pointing equipment. Based on the categorization of indirect input devices presented in [11] we tried to find representative candidates that meets our ergonomics considerations:

- Handheld mouse representing **trackball** technologies.
  Air Mouse with Trackball (Figure 3.2)
- One handed game-controller representing **joystick** technologies.
  Nunchuck from the Wii controller manufactured by Nintendo (Figure 3.3)
- Ring mouse representing **touchpad** technologies.
  Ring mouse 2 manufactured by Genius (Figure 3.4)
- Smartphone representing **multitouch** pad technologies.
  iPhone 4 manufactured by Apple (Figure 3.5)

Figure 3.2: Handheld mouse

Figure 3.3: Game-controller

Figure 3.4: Ring Mouse

Figure 3.5: Smartphone

While some of these devices give us a great freedom in terms of designing their user interface, others have a quite limited amount of inputs; for example the Nunchuck as two buttons, one analog input and an accelerometer while the Smartphone has a

screen that provides the opportunity to layout any type of two dimensional interface, it includes several sensors including an accelerometer and its capacitive screen allows different multitouch gestures.

3.3 Architecture of the system

We created a desktop application in order to combine the information from the eye-tracker and the information from the input devices (Figure 3.6).

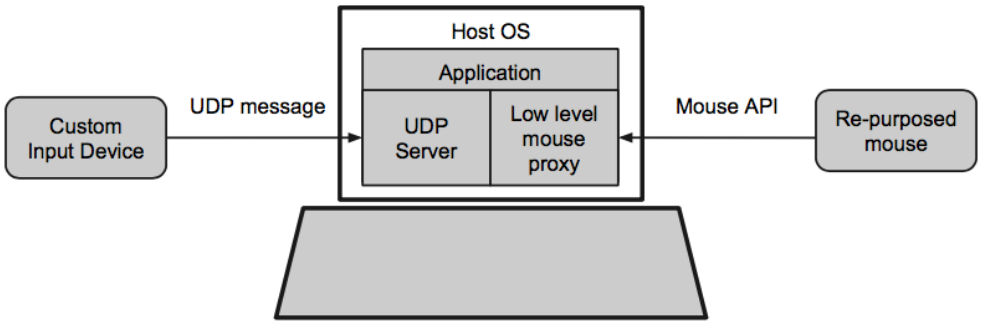

Figure 3.6: Custom input devices uses different protocols that a re-purposed mouse

For this study we are using a Tobii Eye-X Controller that at the time was only compatible with the Microsoft Windows operating system. For this reason, we created a C# native Windows application using the Tobii EyeX Software Development Kit for the Microsoft .NET platform.

The candidates for this study can be classified in custom input devices (smartphone and nunchuck) and re-purposed mice (trackball and ring).

The custom input devices send UDP messages to the desktop application's UDP server. In order to ease the discovery and the configuration of this connection we are using Bonjour, also known as zero-configuration networking, that enables automatic discovery of devices and services on a local network using industry standard IP protocols [1].

The re-purposed mice act like a regular mouse connected to the operating system, by using low level libraries provided by the operating system we can intercept the native mouse events and override or redefine the resulting behavior.

The main responsibilities of the desktop application are:

[1] https://www.apple.com/support/bonjour/

- Advertise the service through Bonjour to be discoverable by custom input devices
- Sense the eye-tracker's in order to get the gaze information
- Receive events from custom input devices in the form of UDP messages
- Receive events from the experiment application in the form of UDP messages
- Override the native mouse events created by the input devices that are repurposed mice
- Control the operating system mouse cursor

The events includes a timestamp, the type of device, and the data needed to interpret that event. The events that a custom device can send are:

- Mouse move with the *(x,y)* delta displacement
- Mouse down and up with button information (i.e. left or right)
- Mouse scroll with the *(y)* delta displacement

These atomic actions can be combine to create more meaningful interactions. A mouse click can be achieved by a mouse down and a consecutive mouse up. A dragging operation is a mouse down, followed by any number of mouse move events and then a final mouse up.

The desktop application saves log information about the position of the gaze, mouse events and also information coming from the experiment application to create a consistent log that was used to obtain the results of the experiment.

3.4 Direct mapping

The companion device should be capable of providing all of the current basic functionalities that any mice have:

- **Pointing:** *(x,y)* delta displacement
- **Click:** left and right click
- **Scrolling:** up and down scroll-wheel
- **Dragging**

While some mice have extra buttons binded to functions like “back page”, for this thesis we only selected the minimum set of operations needed to operate a WIMP interface. Double and triple click are interpret as more than one click and do not need a special treatment.

The companion device should provide an interaction that requires less or equal force for the operation, amount of repetitions and duration of the task than with other devices, it also allows a distant interaction from the computer screen to improve the posture and increase the chances of having a resting position.

Taking into account the duration and repetition aspects in combination with the discoverability of the interface we decided to use a direct mapping approach between the inputs on the companion device and the corresponding mouse operations. The option that was discarded was the use of an non-direct mapping by for example making the companion device an activator of an on-screen menu that provides the desired actions, after opening this menu the user needs to choose one action explicitly.

In order to illustrate the two options imagine the action of scrolling through a document or a website. If the user wants to scroll, the mouse provides a *scroll-wheel* that can be roll up or down.

In the direct mapping approach the companion device provides physical inputs to activate this *scroll up* and *scroll down* actions (e.g. panning over a particular region of the smartphone screen).

In the non-direct mapping approach when the companion device receives the manual input, the host application in the desktop computer opens a menu with the available operations (e.g. scroll up and scroll down) and then the user select the desired operation by giving another physical input on the companion device. The non-direct mapping approach needs not only more than one input but also increases the duration of the task and the repetition of commands by adding the same operation over and over again in order to open the options menu.

Besides the physical ergonomics inconveniences of the non-direct mapping approach, the familiarity with the interface is lost and occlusion problems (i.e. the menu covering an important area of the screen) should be resolved. For all these reasons we decided to have a direct mapping approach trying to avoid any intermediate operations between the physical interaction and the resulting mouse operation on the screen. The direct mapping approach also allow a seamless integration with the current desktop applications without requiring any modification on them.

In the following subsections we are describing the direct mapping for each of the four candidates.

3.5 Explicit versus implicit teleportation

In order to teleport the cursor from its current position to the point of gaze two approaches were considered.

In the explicit approach when the user wants to move the cursor to a particular position he needs to execute two different physical inputs, first he needs to trigger the teleport command and make the cursor "jump" to the estimated point of gaze; after this teleportation the user can refine the position of the cursor with another physical input in order to reach the desired location on the screen.

In the other hand, the implicit approach combines in one physical and continuous action two effects on the cursor: the teleportation and the refinement. As soon as the user triggers a movement in the companion device a teleportation to the current estimated gaze point is executed and any further movement is translated as a refinement of the position of the cursor.

During the design phase we started with the explicit approach but our preliminary tests showed that the approach was difficult to grasp for the users.

3.6 Candidates

3.6.1 Trackball

The hand-held trackball device worked as a regular mouse directly out of the box, it comes with an USB adapter that provides a wireless connection between the device and the computer.

In terms of physical inputs it has two buttons on the upper part, the tracking ball in the middle, three buttons in the lower part (including a scroll-wheel) and a trigger button. The hand grabs the device letting the index finger control the trigger and the thumb can rest on top of the tracking ball or over the scroll-wheel (Figure 3.7).

The mappings for this device are the following:

- Pointing is done with the tracking ball
- Left click is done with the trigger button
- Right click is done with button number four
- Scrolling is done with the scroll-wheel
- Dragging is done by moving the tracking ball while the trigger button is pressed

Figure 3.7: The index finger controls the trigger while the thumb can be used to control the trackball or the scroll-wheel

3.6.2 Joystick

The nunchuck and the Wiimote (Wii Remote) are the two components of the Wii controller manufactured by Nintendo (Figure 3.8).

Figure 3.8: A Wii controller composed by the nunchuck (left) and the Wii Remote (right)

For this study we are using only the nunchuck to be a one handed held lightweight controller. We wanted to connect the nunchuck directly to the computer (instead to the Wiimote). In order to do so we wired the nunchuck to an Arduino board (Figure 3.9) to have access from a Python script to their sensor information.

We loaded a program in the Arduino board to extract the values of the nunchuck's sensors (buttons, analog stick and accelerometer). We created a Python script that reads these values from the Arduino's serial port and sent this information to our

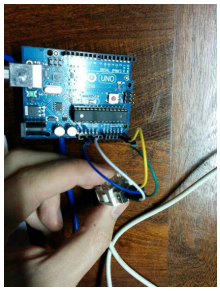

Figure 3.9: The nunchuck is connected to the Arduino board

desktop application through UDP packages, in the same way that the smartphone does but without the need of using a wireless connection.

Our initial mappings (Figure 3.10) for this device were the following:

- Pointing is done with the analog stick
- Left click is done with the "Z" button (the big one)
- Right click is done with the "C" button (the small one)
- Scrolling is done with the analog stick
- Dragging is done by moving the analog stick while the "Z" button is pressed

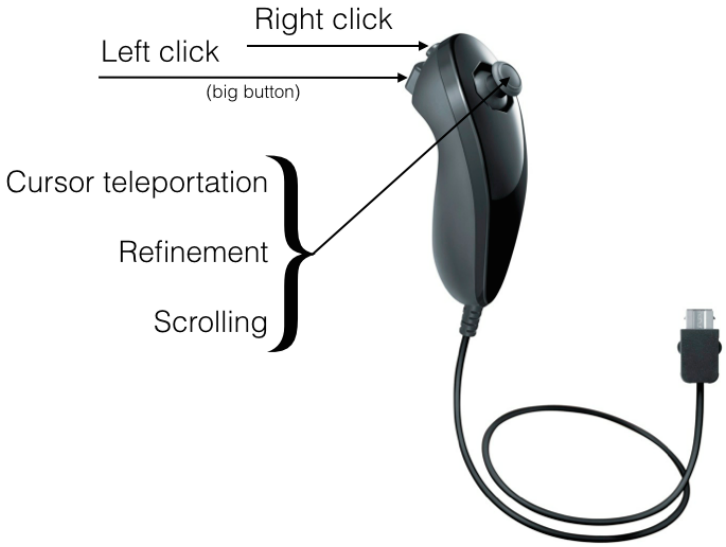

Figure 3.10: The first mappings on the nunchuck before leaving the explicit teleportation approach

Initially when we were considering the explicit teleportation approach we overloaded the analog stick with 3 functions according to the position of the stick. The possible locations of the stick can be graph in a circle, we divided this circle in four concentric regions (Figure 3.11). From inside out: the smallest (first 5%) was the neutral position, the second one (from 5% to 30%) was used for triggering the teleportation explicitly, the third one (from 30% to 95%) to refine the cursor position and the last one (greater than 95%) was used to scroll. Only when the stick reaches the outer most region the scrolling action would be triggered.

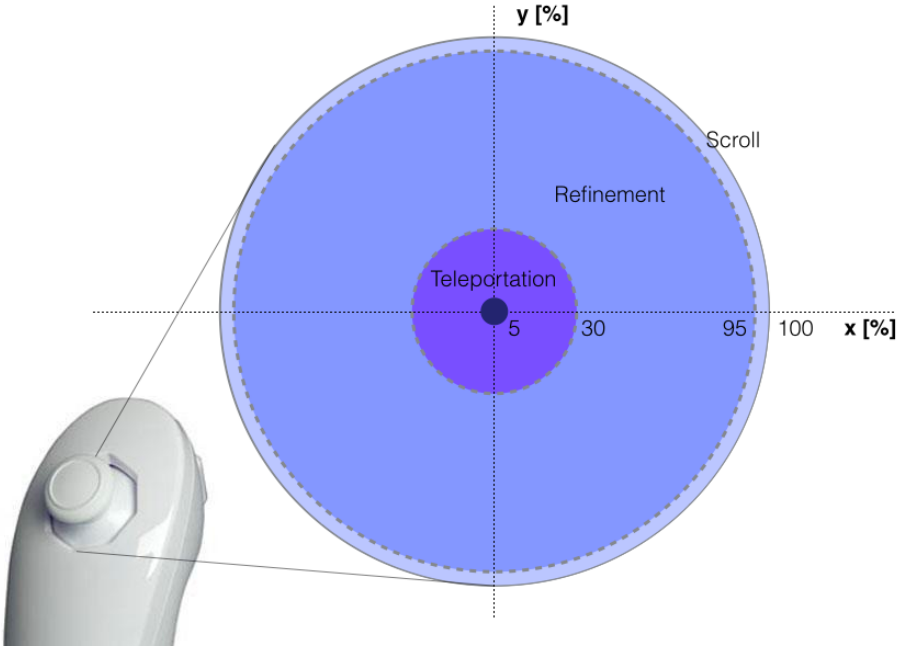

Figure 3.11: The four concentric regions of the analog stick

After our preliminary tests this design was not received positively by the users, because sometimes scrolling was triggered by accident when the user was simply pointing —they tried to move the cursor with the outermost region. Because of that reason we decided to add a scrolling mode (a pseudo mode) by maintaining the "C" button and use the outermost region just as an extension of the third one. When the "C" button is maintained the analog stick movements trigger an scroll action instead of a move action. We also abandon the explicit teleportation approach as explained previously.

The **final mappings** use an implicit teleportation approach combining the second and the third region:

- Pointing is done with the analog stick

- Left click is done with the “Z” button (the big one)
- Right click is done with the “C” button (the small one)
- Scrolling is done with the analog stick while the “C” button is pressed
- Dragging is done by moving the analog stick while the “Z” button is pressed

3.6.3 Ring

Input devices that needs to be held by the user have an associated acquisition time, that is known as the average time to pick up or put down an input device [11]. From the four candidates, three devices are hold: the trackball, the nunchuck and the smartphone. Their weights are 88 grams, 91 grams and 137 grams; respectively. Due to their weight it is not recommended to transform these devices in mounted devices.

In contrast, the ringit is mounted on the user’s hand (Figure 3.12) and the acquisition time is significantly lower in comparison. It is used with the thumb and can be mounted on any of the other fingers like a regular ring; the index and middle finger resulted the more comfortable options.

Figure 3.12: The ring mounted on the user’s index finger

We did not evaluated in deep the duality between held and mounted devices but we see a potential benefit in lightweight devices that can be mounted on the user, this includes the ring used in this study or any other mounted wearable device (e.g. gloves, wrist and armbands).

Our mappings (Figure 3.13) for this device were the following:

- Pointing is done with the rounded touchpad
- Left click is done by pressing the rounded touchpad (there’s a click mechanism below it)
- Right click is done with the right hand size metallic button

- Scrolling is done with the analog stick
- Dragging can be done with the rounded touchpad by click and moving or by
  - Pressing the top centered button (equivalent to mouse down)
  - Using the rounded touchpad to move
  - Releasing the dragged object by click or pressing the top centered button again (equivalent mouse up)

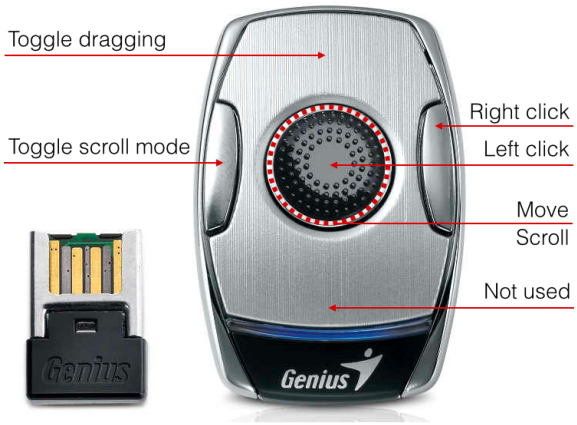

Figure 3.13: The mapping of the ring

Dragging is particularly difficult in this device if is tried to be achieved only by using the rounded touchpad. While it is possible to drag an object with the rounded touchpad, the room for refinement (cursor movement) is small and limited to the size of the touchpad. There is another way to drag an object: first, once the cursor is on top of the desired object one press to the top centered button (Figure 3.13) acts like pressing (and maintain) the left mouse button (mouse down); second, a regular cursor movement can be done (including any teleportation) with the rounded touchpad; finally, only when the top centered button is pressed (mouse up) or a click (left or right) is detected the dragging is stopped.

3.6.4 Smartphone

The states diagrams in this document follow these conventions:

- Each circle represents a state
- The arrow between them is a transition

- A description (text or an image) centered on top of the arrow representing the event that triggers the state transition
- (Optionally) A text in bold and the end of the arrow representing an event that executes before the state transition takes place

The smartphone is the only candidate that does not provide a haptic feedback (to the granular level of mouse action) because it does not have dedicated input controls for the different actions; we are not considering buttons like volumen up, volumen down or on/off button. The great freedom in terms of design that we gain with the screen of the smartphone (e.g. we can layout any type of two dimensional interface, rotate the screen and create any type of finger gesture) has a big drawback—we lose all texture feedback.

For this study we are only considering right handed participants even if all of the prototyped input devices can be used by left handed users, in the case of the smartphone the interface can be configured according to the desired hand but this feature was not implemented in this work.

Another difficulty in mapping a multitouch screen to mouse events is that touch-activated devices do not sense the same two states as the mouse [11] (Figure 3.14).

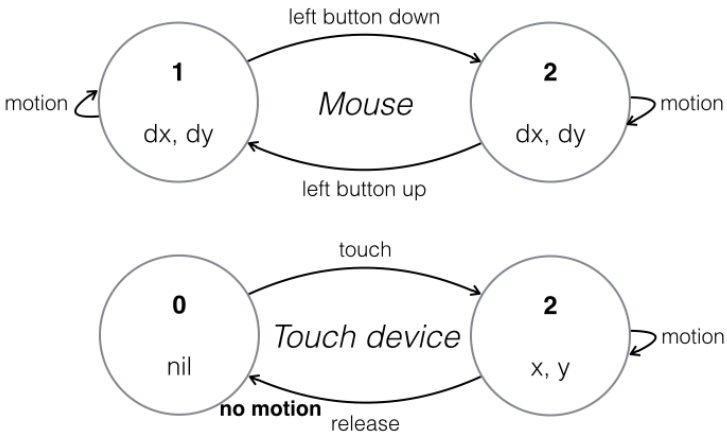

Figure 3.14: States sensed by a mouse versus states sensed by touch-operated devices such as smartphones. Adapted from (Hinckley, Czerwinski & Sinclair, 1998a)

In mainstream computer devices like the MacBook Pro laptop (Figure 3.15), the multi-touchpad (known as "trackpad") not only has buttons but also introduces a new gesture vocabulary to provide the same functionalities than the mouse. For example, if we want to drag an object without using the clickable buttons of the "trackpad" and by only using the touch capabilities of the device, dragging can be done with "tap and drag" or "three fingers drag" techniques. "Tap and drag" works like this: first, the user taps (one finger down and up) the object of interest, if within a small time

threshold (around 300 ms) the user puts one finger down and starts panning (moving the finger on top of the multi-touchpad) the system interprets that the user is doing a dragging; if not the system interprets a left mouse click. To end the dragging there are two modes, with and without "drag lock". Without drag lock, dragging stops immediately after you lift your finger after dragging. With drag lock, dragging continues even if you lift finger and one tap on the trackpad is required to stop. With "three fingers drag" the system uses the amount of fingers as a pseudo mode, when three fingers are detected the movement is interpreted as a dragging. Other amount of fingers can be used, for example two fingers panning on top of the "trackpad" are interpreted as a scroll; other functionalities are mapped to more fingers with other gestures (instead of pan you can swipe, rotate or pinch).

Figure 3.15: A MacBook Pro manufactured by Apple showing the "trackpad" a multi-touchpad used as the pointing device of the machine

On the "trackpad", users are able to interact with all of their fingers. For most common mouse actions the index finger, the middle finger and the thumb are preferred. Index finger is used to point, index and middle finger are used for scrolling and, sometimes, the thumb is used in combination with the index finger (e.g. a drag action by clicking the "trackpad" button with thumb and panning with the index finger). For our prototype, the user is holding the smartphone and this makes the thumb the only finger able to interact with the screen. This eliminate the possibility of using the "three fingers drag" technique or any other interaction method that needs multiple fingers. A big drawback of "tap and drag" is the delay between the first tap and the panning, if the user just wants to do a left click he needs to wait for the time threshold (around 300 ms in the case of the "trackpad") to see the action triggered on the screen.

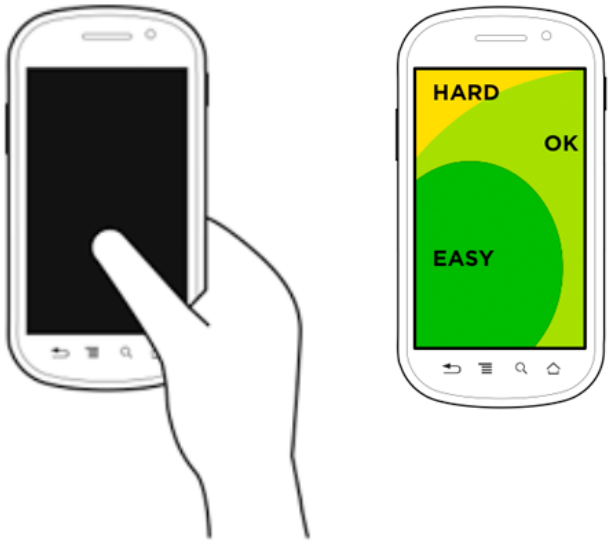

Figure 3.16: Reach of the thumb while holding a device with one hand [28]

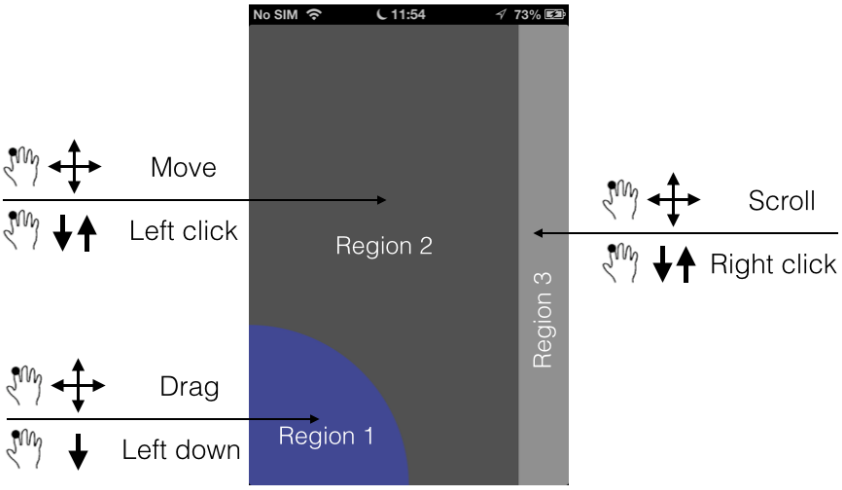

Figure 3.17: The layout and the mappings of the smartphone prototype with their corresponding mappings

When a smartphone is hold with one hand not all of the surface is reachable with the thumb of the hand that holds the device. Several heuristics have been presented (see Figure 3.16 for one example) to show the reach of the thumb. Following this idea we designed an interface that has three regions (Figure 3.17) that activate the different mouse actions and that are accessible with the thumb.

We designed three ways of dragging: the "tap and drag" technique (Figure 3.18), one that uses the region 1 as a pseudo drag mode and the "fat thumb" interaction technique [4].

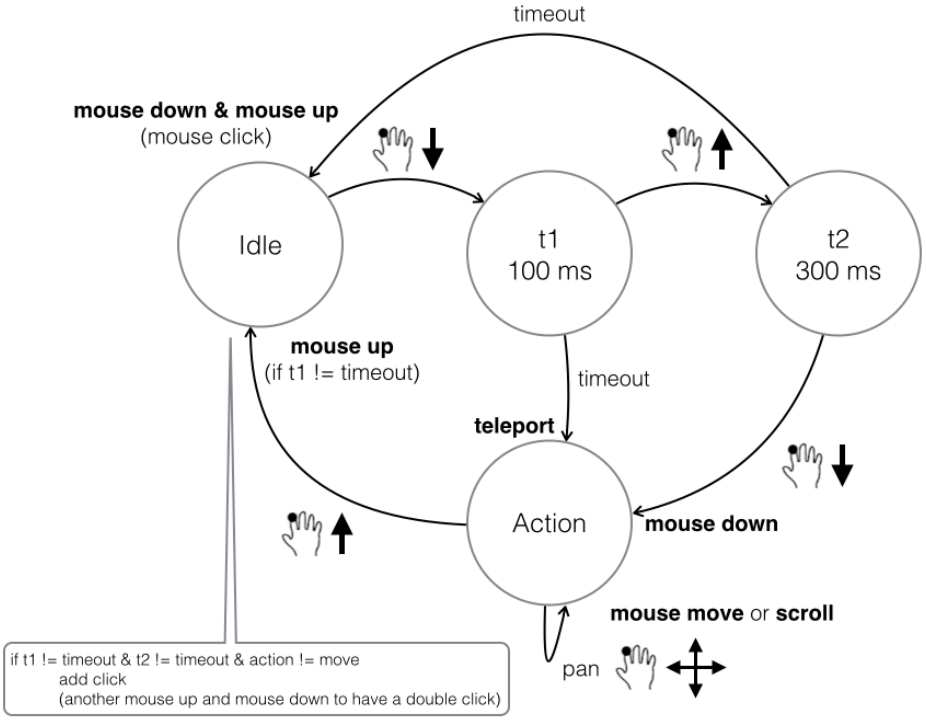

Figure 3.18: In order to implement "tap and drag" two timer states were introduced

In this study we are using the following methapor to justify the use of the "fat thumb" technique for our dragging interaction: imagine that you want to move a lightweight sheet of paper that is on top of your deck with one finger. In order to do so, it could be possible that just gently posing one finger on top of the paper and then moving your hand will not be sufficient to move the paper and in order to move it you will need to put some pressure over it with your finger and then move your hand. Some multi-touchpad technologies allows the developer to access information about not only the position of the touch but also an approximate radius of the touch (other devices can sense pressure but is not as widely available as the radius information). With this radius information we can add a degree of freedom on region 2: if the radius of the touch is small we map the action with a regular mouse move but if the

radius of the touch is big we assume that the user wants to drag and we map the action with a mouse down and then a mouse move.

After our preliminary testings we found that the delay introduced by the "tap and drag" would make our quantifications difficult to compare, for this reason we decided to remove that dragging technique. The "fat thumb" needs a one time calibration per user and it was difficult to discover by the users despite that, some users really like it after some training.

Our mappings (Figure 3.17) for this device are the following:

- Pointing is done by panning (one finger down and move) in region 2
- Left click is done by tapping (one finger down and up) in region 2
- Right click is done by tapping (one finger down and up) in region 3
- Scrolling is done by panning (one finger down and move) in region 3
- Dragging can be done in region 2 using the "fat thumb" on region 2 or by
  - Putting one finger down in region 1 (equivalent to mouse down)
  - And start panning the finger (starting in region 1 but continuing the panning on any region)
  - Releasing the dragged object is done by lifting the finger (equivalent to mouse up)

We extended the functionality of the smartphone when it is rotated to landscape mode by showing a virtual keyboard that allows text input (Figure 3.19). This functionality was not considered in the experiment by disabling the rotation capabilities of the smartphone.

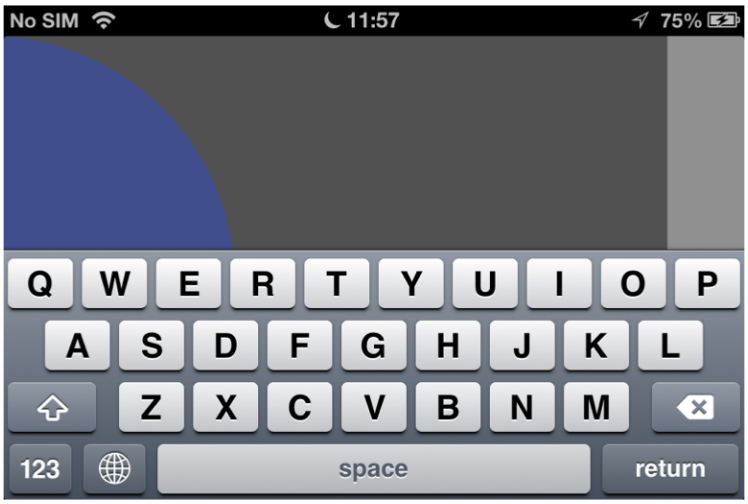

Figure 3.19: We disabled the landscape mode that allowed a keyboard input to focus our study in pointing tasks

Chapter 4
4 Evaluation

In this study we are quantifying the movement time and accuracy of the input device and we are also analyzing the overall user experience with questionnaires.

We conducted a user study with the four prototypes for pointing and dragging tasks. We left outside of the study the scrolling interaction because its behavior was not augmented with the gaze information.

4.1 Method

We used a within-subjects design with five input devices: a regular optical mouse (manufactured by Dell, model R41108) and the four prototyped devices (nunchuck, ring, smartphone and hand-held trackball).

15 participants (10 male, 5 female) between 23 and 40 years old (M = 27.33, SD = 2.12) from the KTH Kista Campus participated in the experiment. All the subjects were right handed.

Tasks were performed on a MacBook Air (late 2013) running the desktop application on a Windows operating system. The mouse sensitivity was set in the default value and the mouse acceleration was removed but each device had his own sensitivity inherent to their physical properties, for example the nunchuck stick is slow and the tracking ball of the trackball is fast when compared with other devices. The visual output of the laptop was connected to an LCD screen in a 1680 x 1050 pixels resolution setup.

With each of the five input devices two task blocks had to be completed in the same order, first B_p (pointing) and then B_d (drag and dropping). The pointing devices were counterbalanced to prevent the influence of order effect but the mouse was always selected as the first device to introduce users to the eye tracking capabilities. Each block has sixteen runs based on the combination of four distances (D = 1, 2, 3 and 4 being 128, 256, 512, 1024 pixels respectively) and four target sizes (S = 1, 2, 3

and 4 being 16, 32, 64, 128 pixels respectively). Prior to each new block condition, subjects were given a practice block in order to get used to the new device-task and only when they decided to be ready the real task block started.

4.2 Procedure

To study the pointing and dragging tasks we used a similar approach to the one presented in [19]: they compared the performance during pointing (State 1) and dragging (State 2) task of three devices: a mouse, a stylus and a regular trackball.

The targets used in our experiment are located randomly on the screen, their size is one of four available sizes and the distance that the cursor needs to travel is selected from four available distances. During the pointing task the distance of interest is between the mouse cursor before moving and the center of the target. During the dragging task the distance of interest is between the center of the *draggable object* and the *drop area*; the distance from the mouse cursor and the *draggable object* is not of our interest because it is studied on the pointing task.

Each pointing task block consists on hitting one of the sixteen targets at a time with one device (Figure 4.1). When the user positions the cursor on top of the target, the target gets highlighted (Figure 4.2). Once the cursor is on top of the target and the user triggers a click event with the input device, the current target goes away and the next one appears.

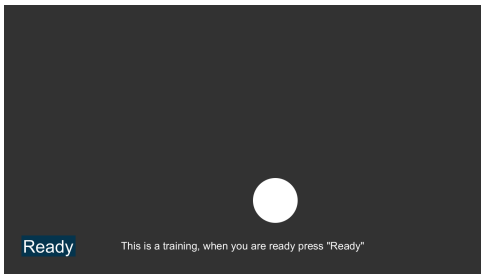

Figure 4.1: A target is represented as a white circle

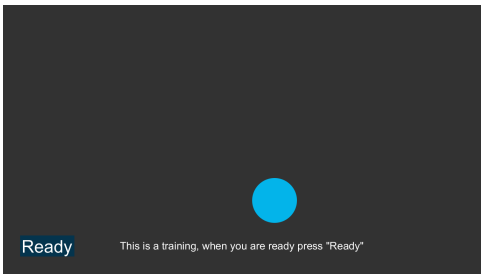

Figure 4.2: When the cursor is on top of the target, the target is highlighted

For the dragging task, we are using a similar approach but two targets appears on the screen: a *draggable object* identical to the target on the pointing task and a *dropping area* represented with a slightly bigger dotted circle (Figure 4.3). The *draggable object* has the same highlight effect than the target on the pointing task but when it starts being dragged it gets highlighted with a slightly darker tint color. Once the *draggable object* is on top of the *dropping area*, the area is highlighted and a mouse release completes the task (Figure 4.4).

Figure 4.3: Draggable object as a circle and drop area as a dotted circle

Figure 4.4: The drop area is highlighted when the target is on top of it

4.3 Measures

Our logs included target acquisition time (movement time), accuracy and error rate (releasing the dragged object outside the drop area or clicking outside of a target is considered as an error). The session ended with a questionnaire handed to the subject to rate the different devices using a 5-point Likert scale for the following characteristics: ease of use, ease of learning, apparent speed and accuracy for both tasks, strain on the muscles of the hand and overall satisfaction.

Due to the high variability on the data we could not perform a statistical test (e.g ANOVA). We can see the movement time data in a boxplot for the pointing task (Figure 4.5) as well for the drag and dropping task (Figure 4.6). Any data point that is more than 1.5 box-lengths up to 3 box-lengths from the edge of their box is classified as an outlier and is represented with a dot; if it is more than 3 box-lengths it is represented as a star.

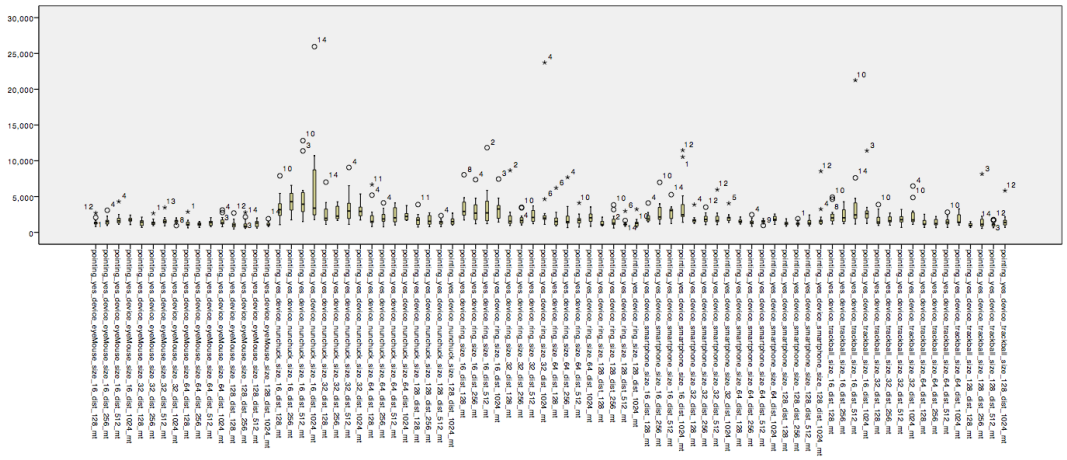

Figure 4.5: Movement time (ms) data represented in a boxplot for all the sizes and distances combination for the pointing task. Dots and stars are outliers

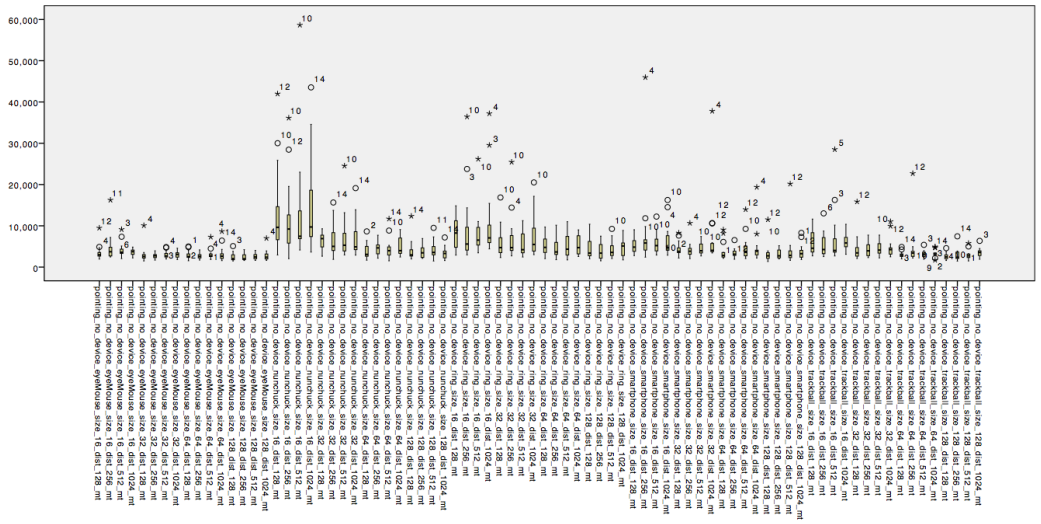

Figure 4.6: Movement time (ms) data represented in a boxplot for all the sizes and distances combination for the drag and dropping task. Dots and stars are outliers

The presence of these outliers can be attributed to three main reasons: eye-tracker calibration issues and inaccuracies, low amount of participants and unfamiliarity of the participants with the use of the eyes for these type of tasks.

4.3.1 Pointing

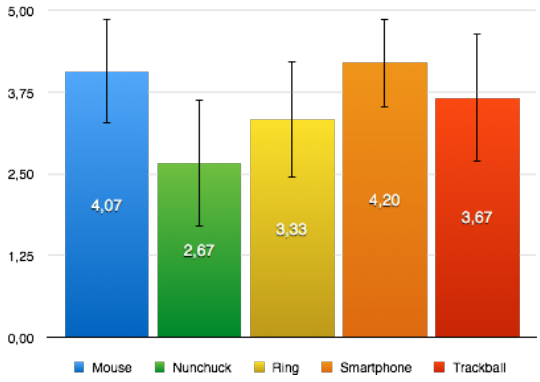

Figure 4.7: Pointing speed perceived by the participants for each device (the higher the better)

Users perceived the smartphone as the fastest device for pointing even when comparing it with the mouse (Figure 4.7). When comparing the results with the measured movement time for all the devices (Figure 4.8) the mouse is by far the fastest device,

followed by the smartphone and the trackball.

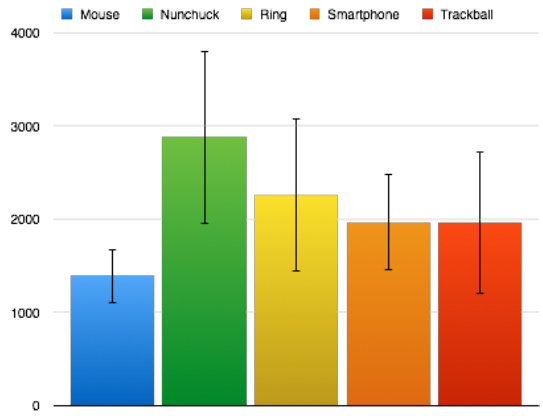

Figure 4.8: Comparison of mean movement time (in ms, the lower the better) during the pointing task across all the devices

Analyzing the measured movement time for the pointing task between the mouse (Figure 4.9) and the smartphone (Figure 4.10) clearly shows that the mouse is faster in all of the sizes and distances combinations —being the difference highly noticeable for small targets (e.g. $size_1$).

In summary, for **pointing speed** the perceived speed was aligned with the measured movement time, with the exception of the mouse that quantitatively was not in the same category than the smartphone.

We can see in Figure 4.11 the accuracy perceived by the participants and in Figure 4.12 the measured distance in pixels from the cursor position (right after successfully clicking on the target) to the center of the destination target.

For the **pointing accuracy** the perceived accuracy was aligned with the measured accuracy making the smarpthone the most accurate device.

The error rate for the pointing task with all devices ranged between 0 and 10 for all the tests (Figure 4.13). We can notice that the smartphone is the device with the highest error rate.

Two participants underperformed using the smartphone and made 7 and 10 errors with the smallest targets, while the rest of the participants got values between 0 and 5. The high error rate can be explained by this: in order to move the cursor, the finger needs to move over the smartphone's screen to the desired position, and then he needs to lift his finger for doing a click; that lift sometimes causes a small cursor displacement, and when the click is finally done it hits outside the target.

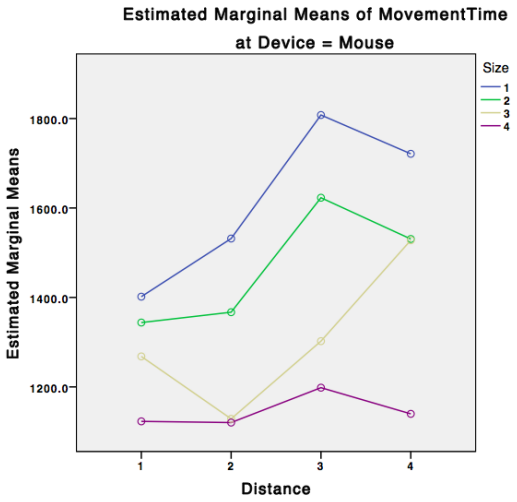

Figure 4.9: Mean movement time (ms) of the mouse during the pointing task showing all the sizes and distances combinations

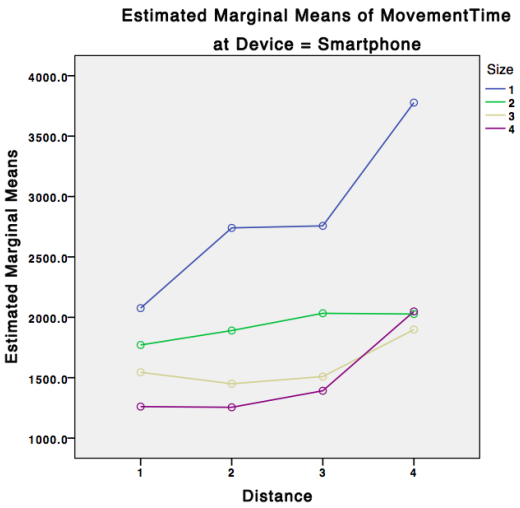

Figure 4.10: Mean movement time (ms) of the smartphone during the pointing task showing all the sizes and distances combinations

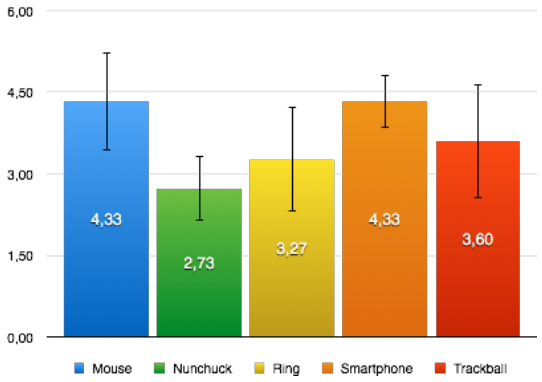

Figure 4.11: Pointing precision perceived by the participants for each device (the higher the better)

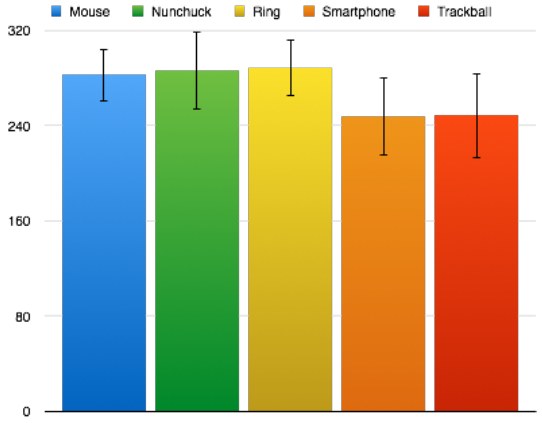

Figure 4.12: Comparison of mean distance from cursor to the center of the target (in pixels) during the pointing task across all the devices. Low values mean higher accuracy

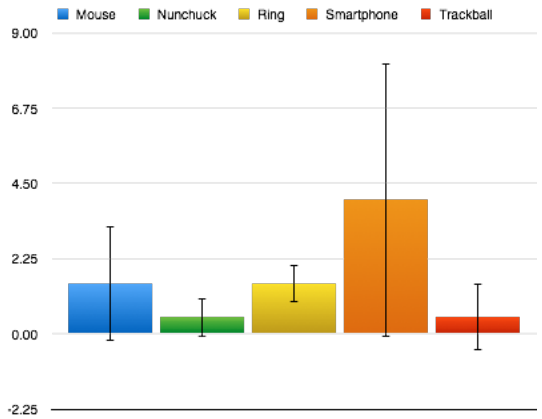

Figure 4.13: Comparison of mean error count during the pointing task across all the devices

4.3.2 Drag and dropping

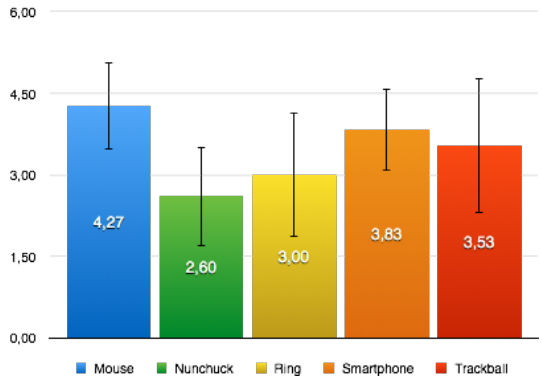

Figure 4.14: Dropping speed perceived by the participants for each device (the higher the better)

The mouse ranked first in the perceived speed during drag and drop tasks (Figure 4.14). From the four prototyped devices, users perceived the smartphone as the fastest device for drag and dropping. When comparing the results with the measured movement time (Figure 4.15) the mouse is still ranking first but from the prototyped devices the trackball ranked better than the smartphone for a small difference. This could be attributed to the high sensitivity that the tracking ball of the prototyped device had.

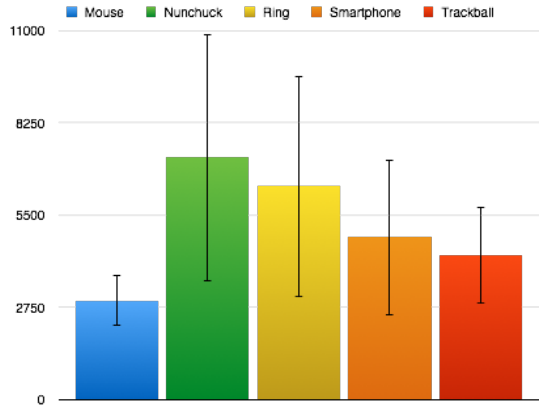

Figure 4.15: Comparison of mean movement time (in ms, the lower the better) during the drag and drop task across all the devices

In summary, for **drag and drop speed** according to the perceived speed, the devices rank as: mouse, smartphone, trackball, ring and nunchuck. For the measured movement time the ranking is: mouse, trackball, smartphone, ring and nunchuck. In general terms both rankings are similar due to the fact that the mouse and the

trackball got almost equal movement times means.

In (Figure 4.16) we can see the accuracy perceived by the participants and in (Figure 4.12) the measured distance of the cursor (after dropping) from the center of destination target (in pixels).

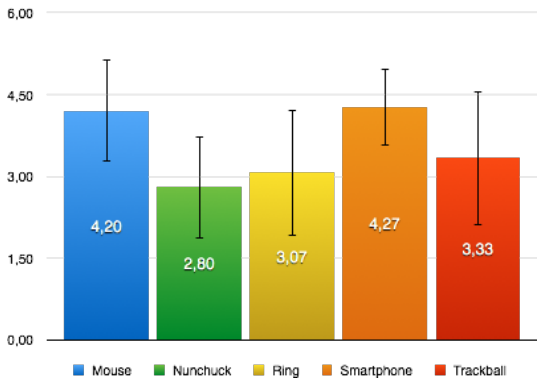

Figure 4.16: Drag and drop precision perceived by the participants for each device (the higher the better)

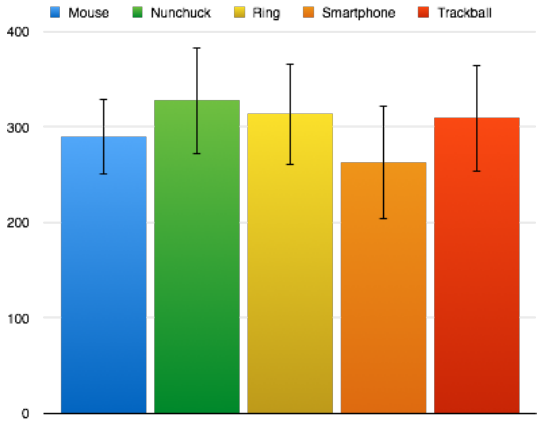

Figure 4.17: Comparison of mean distance from cursor to the center of the target (in pixels) during the drag and drop task across all the devices. Low values mean higher accuracy

In summary, for **drag and drop accuracy** the perceived accuracy was aligned with the measured accuracy.

The drag and drop task has a pointing part (acquiring the target) and the proper drag and drop part. We summarized the error rate during both the target acquisition and the drag and drop parts (Figure 4.18).

Surprisingly, the trackball was the device with the lower error rate (with the mouse

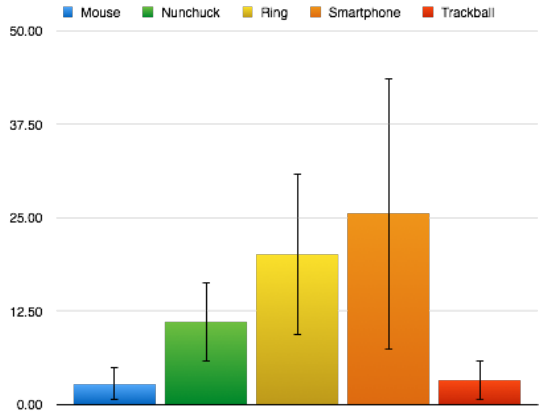

Figure 4.18: Comparison of mean error count during the drag and drop task across all the devices

as a close second). Again, the smartphone ranked first as the device with more errors. The same explanation than we gave for the pointing task applied here but during the dragging part another common error was detected. Because of the lack of physical texture on the screen of the device, participants missed the dragging area several times and they needed to look at the screen of the device to see if they were pressing in the right area.

4.3.3 Overall Satisfaction

In terms of usability, we are evaluating the ease of learning (Figure 4.19) and ease of use (Figure 4.20) perceived by the participants.

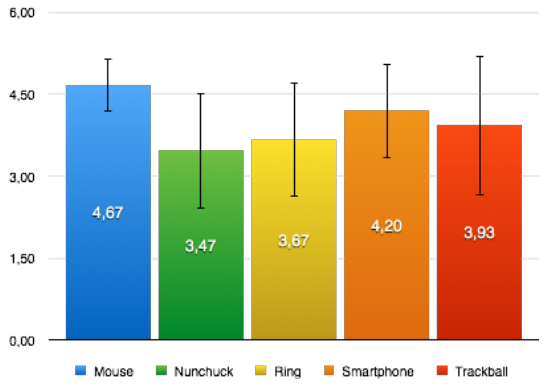

Figure 4.19: Ease of learning

Taking into account only the four prototyped devices, the ranking on usability is: smartphone, trackball, ring and nunchuck. The nunchuck was described as strange or

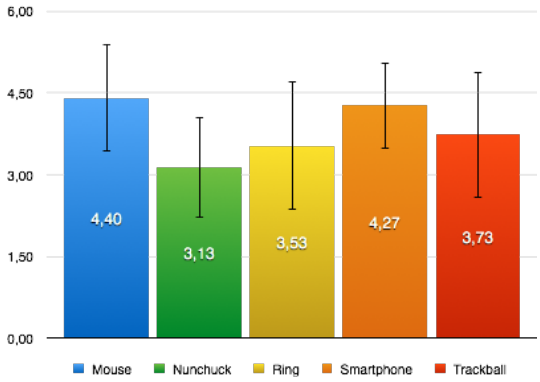

Figure 4.20: Ease of use

slow by several participants, and some of them complained about not being "gamers" while they were exposed to it. The ring was received enthusiastically by all the participants but during the dragging task complications started to appear. The smartphone and the trackball ranked firsts in terms of usability.

We also asked to the participants to rank the devices according to their perceived comfortability, explained to them as how well does the device adjust to your grip and the position of your fingers.

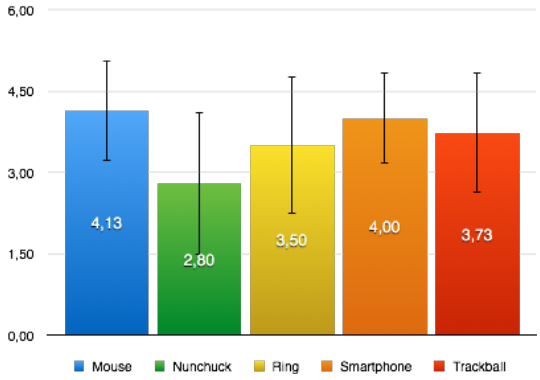

Figure 4.21: Comfortability

We asked to the participants to give an "overall satisfaction" mark (Figure 4.11) to the devices and finally, we asked them to say which device was their favorite from the four prototyped devices. The "favorites rank" was: smartphone (9 votes), trackball (4 votes), ring (2 votes) and nunchuck (0 votes). The "overall satisfaction" mark and the "favorites rank" gave the same device ranking.

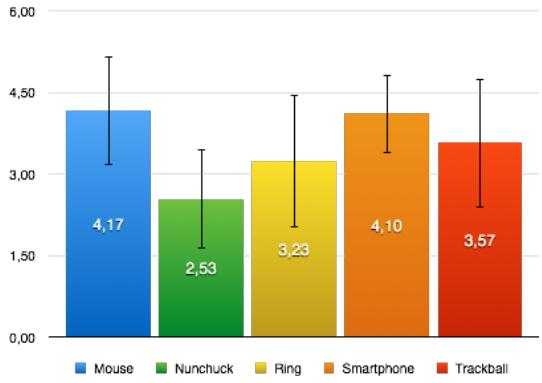

Figure 4.22: Overall satisfaction

Chapter 5
5 Conclusion

From the results of our study we can see that the prototype that provides the best overall satisfaction is the smartphone. In terms of ergonomic experience, the smartphone also ranked first. To understand the productivity of the prototypes it is interesting to analyze not only the measures but also the user feedback (Figure 5.1).

	Pointing		Drag and Drop	
	Speed	**Accuracy**	**Speed**	**Accuracy**
Perceived	Smartphone	Smartphone	Smartphone	Smartphone
Measured	Trackball/Smartphone	Smartphone	Trackball	Smartphone

Figure 5.1: Final device ranking for perceived and measured values

From the prototyped devices the smartphone and the trackball were the preferred ones. Interestingly, the measures showed similar results, so the perception of the user was aligned with the real productivity.

In terms of comfortability, the smartphone also ranked first.

Going back to the original questions: *Which prototype provides the highest productivity (in terms of low movement time and high accuracy)?*. The smartphone first and the trackball as a close second.

Which pointing input technology provides the highest productivity and the highest comfort when is used in combination with eye-tracking technologies?. We cannot answer this question because we cannot generalize the result of this experiment to all the usages of the input technologies but particularly for this experiment the multitouch technologies and the trackball technologies provided the highest productivity and comfort.

5.1 Discussion

Even if participants where instructed to maintain a distant position from the desktop, more than a half of the participants used the desk table as a support for holding their hand while they were grabbing the device (Figure 5.2 and Figure 5.3). Sometimes, the opposite happened: relaxed participants tended to move too apart from the computer screen, making the eye-tracker lose their gaze information.

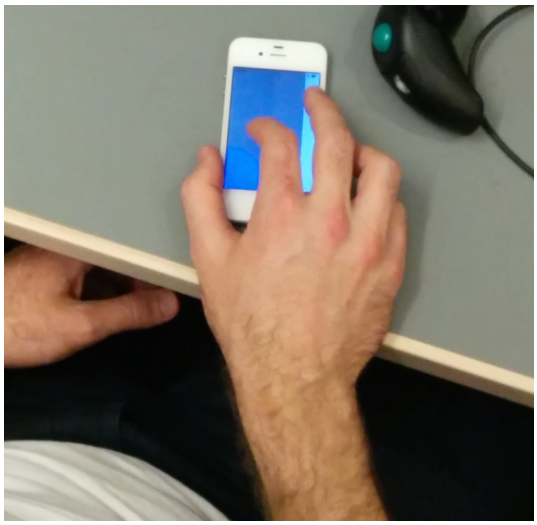

Figure 5.2: A participant that is using the desk to hold the smartphone

Another of our hypothesis was that the device should allow one handed interaction, but because the participants lacked any ergonomic problem they tended to use both hands to interact with the device (Figure 5.4 and Figure 5.5) and some of them even used their non-dominant hand to support their dominant arm (Figure 5.6).

Figure 5.3: A participant using the desk to support his arm while using the nunchuck

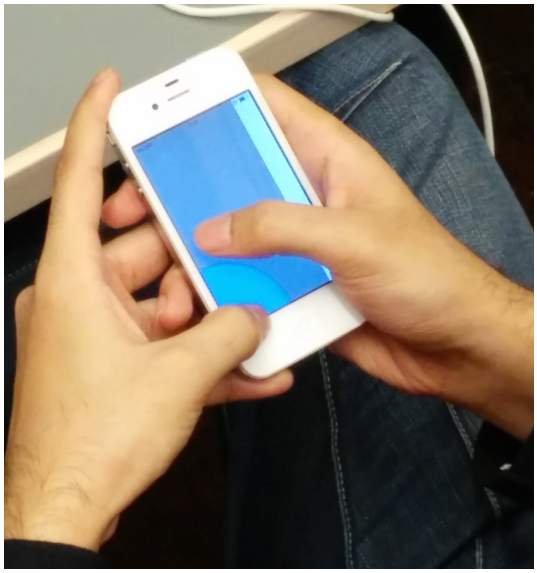

Figure 5.4: A participant dragging with their non-dominant hand and moving with their dominant hand while using the smartphone

Figure 5.5: A participant supporting the trackball with his leg while hold the device with two hands

Figure 5.6: A participant supporting their dominant arm with their non dominant arm while using the trackball

Chapter 6
6 Further work

During this study we compared four gaze pointing devices on two specific tasks (pointing and drag and drop).

One natural extension of the work presented here is to include measures of the electromyographic (EMG) activity of the muscles in the evaluation; this could provide more concrete information about the physical ergonomics of the devices during the proposed tasks.

While our hypothesis was to design a device that can be used to one hand, our experiments showed that users tend to seek support from the desk or the non-dominant hand while holding the devices. A new direction for future studies could be to start the design process with a bi-manual interaction approach.

More research should be conducted with complex tasks in order to understand the real behavior of users respect to these input technologies combined with eye-tracking—for example web browsing or text editing. We believe that during these more complex tasks users could combine pointing, dragging, dropping, scrolling, keyboard shortcuts and text input in unexpected ways, leaving room for new designs in terms of gaze input devices. As one example, we added an input text function with a virtual keyboard to the smartphone's screen prototype when the device is in landscape mode (Fig 6.1). This work can be extended by understanding how these input devices can extend their functionalities beyond pointing in order to include text input or common shortcuts executions (like copy and paste).

A setup where the user is sitting in a sofa interacting with a large screen positioned two or three meters away was not possible to test with the current remote eye tracking technologies, as they have only a distance range of about one meter. While the range of eye-trackers will improve in the future, an experiment with a head-mounted eye-tracker could be used to test these devices in a setting with longer distances between the user and the computer screen.

In terms of accuracy, the gaze data provided by the eye-tracker always had an offset with the real gaze point of the user. We believe that all the user's manual physical movement used to refine the cursor's position can be used to generate an automatic

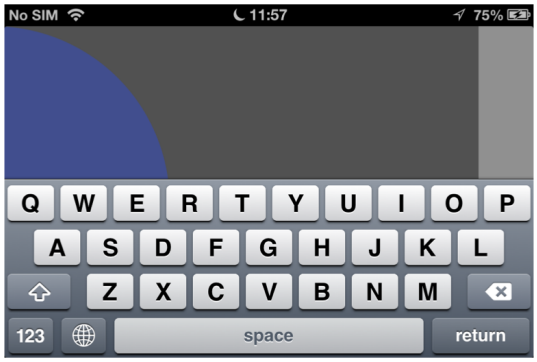

Figure 6.1: An example of how the input devices can handle text input and not only pointing tasks

refinement of the gaze data. One approach could be the one proposed in [9] called Dynamic Local Calibrations; every time that a user manually corrects the cursor's position it is helping the system to generate a vector field that accumulates all of these refinements. The screen is divided in several regions and each region has its own vector field. Once there is enough data for a particular region, the gaze information of that region is corrected with the vector field and then is corrected with the manual input, if necessary. We believe that this approach can reduce the presence of offsets between the gaze data and the real gaze point and at the same time it could reduce the physical movement of the user's hand over time. Also, we detected a lot of overshooting during the experiment. An approach similar to [8] can be used on the prototypes to reduce the cursor's sensitivity near the teleportation position; in this could be possible to maintain a target agnostic approach but increasing the accuracy during the refinement phase. Combining the "Dynamic Local Calibrations" and the "Dynamic Cursor Sensitivity" with the prototypes could increase the ergonomics benefits of the devices.

The main drawback of the smartphone when compared with the other devices is the lack of haptic feedback. Depicting different regions on the smartphone's screen makes the interface dynamic, but users lost track on whether their fingers are in or out of the desired region because of being too focused on the computer screen while performing the tasks. To correct this problem, users tend to stop looking at the computer screen and focus their eyes on the smartphone's screen; this "eyes context switch" reduces their productivity during the task. We prototyped a potential solution for this problem that we called "sticky texture". By using simple paper sticky notes we can add a texture difference between regions. Figure 6.2 shows a "sticky texture" on top of the dragging and the scrolling region. The sticky note does not block the touch detection of the capacitive smartphone's screen but provides a simple way of differentiating the texture of the three regions while fingers are sliding on top of the screen. This is just and idea of a low prototyping technique that needs to be tested, and more research could be conducted in the areas of industrial and

material design.

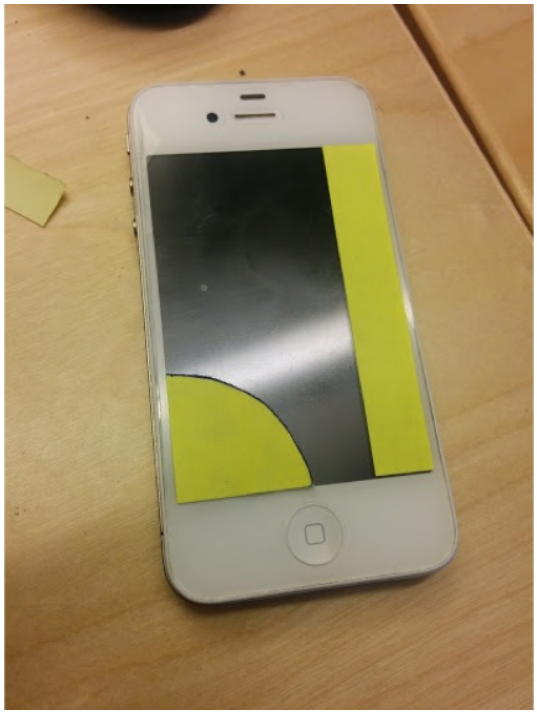

Figure 6.2: "Sticky textures", an example of how we can add haptic feedback to the smartphone's screen

Bibliography

[1] Waseem Amir Bashir, Tetsuya Torio, Francis William Smith, Keisuke Takahashi, and Malcolm Pope. The way you sit will never be the same! alterations of lumbosacral curvature and intervertebral disc morphology in normal subjects in variable sitting positions using whole-body positional MRI. *Radiological Society of North America 2006 Scientific Assembly and Annual Meeting*, November 2006.

[2] Richard Bates and Howell Istance. Zooming interfaces!: Enhancing the performance of eye controlled pointing devices. In *Proceedings of the Fifth International ACM Conference on Assistive Technologies*, Assets '02, pages 119–126, New York, NY, USA, 2002. ACM.

[3] Richard A. Bolt. Eyes at the interface. In *Proceedings of the 1982 Conference on Human Factors in Computing Systems*, CHI '82, pages 360–362, New York, NY, USA, 1982. ACM.

[4] Sebastian Boring, David Ledo, Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. The fat thumb: Using the thumb's contact size for single-handed mobile interaction. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '12, pages 39–48, New York, NY, USA, 2012. ACM.

[5] R. S. Bridger. *Introduction to Ergonomics, Third Edition*. CRC Press, Boca Raton, 3 edition edition, August 2008.

[6] P Brooks. Repetitive strain injury. *BMJ : British Medical Journal*, 307(6915):1298, November 1993.

[7] Andrew Duchowski. *Eye Tracking Methodology: Theory and Practice*. Springer, London, 2nd edition edition, August 2007.

[8] Ribel Fares, Dustin Downing, and Oleg Komogortsev. Magic-sense: Dynamic cursor sensitivity-based magic pointing. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 2489–2494, New York, NY, USA, 2012. ACM.

[9] Ribel Fares, Shaomin Fang, and Oleg Komogortsev. Can we beat the mouse with MAGIC? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1387–1390, New York, NY, USA, 2013. ACM.

[10] Kenneth Holmqvist, Marcus Nystrom, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, Joost van de Weijer, and & 2 more. *Eye Tracking: A comprehensive guide to methods and measures*. Oxford University Press, USA, Oxford ; New York, 1 edition edition, November 2011.

[11] Julie A. Jacko. *Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, Third Edition*. CRC Press, Boca Raton, FL, 3 edition edition, May 2012.

[12] Robert J. K. Jacob. What you look at is what you get: Eye movement-based interaction techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 11–18, New York, NY, USA, 1990. ACM.

[13] Robert J. K. Jacob. The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Trans. Inf. Syst.*, 9(2):152–169, April 1991.

[14] Robert J. K. Jacob. Eye movement-based human-computer interaction techniques: Toward non-command interfaces. In *IN ADVANCES IN HUMAN-COMPUTER INTERACTION*, pages 151–190. Ablex Publishing Co, 1993.

[15] Oha K, Viljasoo V, and Merisalu E. Prevalence of musculoskeletal disorders, assessment of parametres of muscle tone and health status among office workers, 2010.

[16] Daniel J. Liebling and Susan T. Dumais. Gaze and mouse coordination in everyday work. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, pages 1141–1150, New York, NY, USA, 2014. ACM.

[17] James S. Lipscomb and Michael E. Pique. Analog input device physical characteristics. *SIGCHI Bull.*, 25(3):40–45, July 1993.

[18] Cecil Lozano, Devin Jindrich, and Kanav Kahol. The impact on musculoskeletal system during multitouch tablet interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 825–828, New York, NY, USA, 2011. ACM.

[19] I. Scott MacKenzie, Abigail Sellen, and William A. S. Buxton. A comparison of input devices in element pointing and dragging tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, pages 161–166, New York, NY, USA, 1991. ACM.

[20] Frances E. Mount, Mihriban Whitmore, and Sheryl L. Stealey. Evaluation of neutral body posture on shuttle mission STS-57 (SPACEHAB-1). revision. Technical report, February 2003.

[21] Amir Hossein Habibi Onsorodi. *The Impact of Laptop and Desktop Computer Workstation on Human Performance*. Thesis, Eastern Mediterranean University (EMU), 2011. Master of Science in Industrial Engineering. Thesis (M.S.)– Eastern Mediterranean University, Faculty of Engineering, Dept. of Industrial Engineering, 2011. Supervisor: Assist. Prof. Dr. Orhan Korhan.

[22] Ronan O'Rahilly and Fabiola MÃijller. *Basic human anatomy: a regional study of human structure*. Saunders, 1983.

[23] Linda E. Sibert and Robert J. K. Jacob. Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 281–288, New York, NY, USA, 2000. ACM.

[24] Sophie Stellmach and Raimund Dachselt. Look & touch: Gaze-supported target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2981–2990, New York, NY, USA, 2012. ACM.

[25] Jannique G. Z. van Uffelen, Jason Wong, Josephine Y. Chau, Hidde P. van der Ploeg, Ingrid Riphagen, Nicholas D. Gilson, Nicola W. Burton, Genevieve N. Healy, Alicia A. Thorp, Bronwyn K. Clark, Paul A. Gardiner, David W. Dunstan, Adrian Bauman, Neville Owen, and Wendy J. Brown. Occupational sitting and health risks: A systematic review. *American Journal of Preventive Medicine*, 39(4):379–388, October 2010.

[26] Colin Ware and Harutune H. Mikaelian. An evaluation of an eye tracker as a device for computer input. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, CHI '87, pages 183–188, New York, NY, USA, 1987. ACM.

[27] Nicholas Warren and Timothy F. Morse. Neutral posture.

[28] Luke Wroblewski. Responsive navigation: Optimizing for touch across devices.

[29] Pat Wyatt, Kim Todd, and Tabatha Verbick. Oh, my aching laptop: Expanding the boundaries of campus computing ergonomics. In *Proceedings of the 34th Annual ACM SIGUCCS Fall Conference: Expanding the Boundaries*, SIGUCCS '06, pages 431–439, New York, NY, USA, 2006. ACM.

[30] Shumin Zhai, Carlos Morimoto, and Steven Ihde. Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 246–253, New York, NY, USA, 1999. ACM.

Declaration

I hereby certify that I have written this thesis independently and have only used the specified sources and resources indicated in the bibliography.

Stockholm, 1. November 2014

. . . . . .

My Name